IBM

Infrastructure Solutions: Building a Smart Bank Operating Environment

- Using a live showcase to demonstrate how to deliver key banking projects
- Building an SOA with IBM System z and IBM BladeCenter with IBM z/OS and Linux
- Using IBM Tivoli Monitoring V6 and Tivoli Composite Application Manager V6

Christian Matthys
Guillaume Arnould
Neil Ashworth
Stefano Delle Chiaie
Eric Devezeaud
Fabrice Jarassat
Francois Lebe
Detlef Wolf

Redbooks

ibm.com/redbooks

International Technical Support Organization

Infrastructure Solutions: Building a Smart Bank Operating Environment

October 2006

SG24-7113-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (October 2006)

This edition applies to the IBM Smart Bank showcase that demonstrates the value of IBM infrastructure in the retail banking context. In particular, the following IBM products are used for the showcase: for the hardware, IBM System z and IBM BladeCenter; for the middleware, IBM WebSphere Application Server V5; for systems management, IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6.

© Copyright International Business Machines Corporation 2006. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices vii
Trademarks viii

Preface ix
The team that wrote this IBM Redbook x
Become a published author xii
Comments welcome. xii

Chapter 1. Introducing the technical architecture of the Smart Bank showcase 1
1.1 Understanding the Smart Bank showcase 2
1.1.1 Overview of IBM infrastructure solutions. 2
1.1.2 Objective of the Smart Bank showcase 4
1.1.3 The banking sector's current key business drivers 5
1.1.4 The banking sector's future key business drivers 7
1.1.5 The way the IBM Financial Services Sector addresses these issues 8
1.1.6 Demonstration proof points: A day in the life of a bank. 10
1.2 Architectural choices 14
1.2.1 Banking clients' current architecture 14
1.2.2 Reference architectures and models 14
1.2.3 Functional requirements 15
1.2.4 Nonfunctional requirements 16
1.2.5 Key architectural decisions 16
1.3 The architectural components 17
1.3.1 The current architecture of the Smart Bank showcase 18
1.3.2 Systems management requirements 22
1.3.3 Use cases this IBM Redbook focuses on 23
1.4 IBM Tivoli Monitoring: Architectural design and description 23
1.4.1 Products selected from the Tivoli portfolio 24
1.4.2 General architectural considerations. 32
1.4.3 Distributed environment scenario 35
1.4.4 IBM System z environment scenario. 37
1.4.5 Distributed systems and IBM System z working together 44
1.4.6 Virtualized IBM System x with Moveable Hub implementation 46
1.5 Conclusion 49

Chapter 2. IT environment of the Smart Bank showcase 51
2.1 The IBM System z environment 53
2.1.1 The hardware components 53
2.1.2 The operating systems and the middleware components. 54
2.1.3 IBM Tivoli components 57
2.1.4 Summary. 60
2.1.5 Configuring the Smart Bank showcase 60
2.1.6 OMEGAMON XE Tivoli enterprise management: Initialization checklist for the server and the agent 81
2.2 The distributed environment 84
2.2.1 Overview of the components 84
2.2.2 Platform description 85
2.2.3 Overview of network topology 86
2.2.4 Network communication 87

© Copyright IBM Corp. 2006. All rights reserved.

2.2.5 Estimating the capacity of the monitoring environment. 90
2.2.6 Failover support 91
2.2.7 Operating system-specific considerations. 92
2.2.8 Database considerations 94
2.2.9 Installing the monitoring servers 95
2.2.10 Deploying the remote operating system agents 104
2.2.11 An alternative solution. 107
2.2.12 Installing the key components for the distributed environment 108

Chapter 3. Using Physical views 121
3.1 Physical views from IBM Tivoli Monitoring V6. 122
3.1.1 IBM OMEGAMON for z/OS V3.10 125
3.1.2 IBM OMEGAMON for UNIX System Services V2.20 127
3.1.3 IBM OMEGAMON for DB2 V3.10 128
3.1.4 IBM OMEGAMON for Mainframe Networks V3.10 129
3.1.5 IBM OMEGAMON for CICS V3.10 130
3.1.6 IBM OMEGAMON for Storage on z/OS V3.1.0. 131
3.2 Physical views from IBM Tivoli Composite Application Manager 131
3.2.1 IBM Tivoli Composite Application Manager for Service-Oriented Architecture . . 132
3.2.2 IBM OMEGAMON for WebSphere Application Server 139
3.2.3 IBM OMEGAMON for WebSphere MQ. 149
3.3 IBM CICS Business Event Publisher for MQ Series and event processing 154
3.3.1 Overview of CICS Business Event Publisher for MQ Series. 154
3.3.2 Using CICS Business Event Publisher to populate the banking data warehouse database. 157
3.3.3 Monitoring a running CICS Business Event Publisher using OMEGAMON XE for MQSeries 161
3.4 Multichannel architecture: Physical views. 170
3.4.1 WebSphere Application Server. 170
3.4.2 Uniform Resource Locator monitoring 171
3.4.3 IBM HTTP Server monitoring 173
3.4.4 Simple network monitoring 177

Chapter 4. Customizing the Logical views 181
4.1 Introducing the Logical view 182
4.1.1 Things to be considered before defining the Logical view 182
4.1.2 Defining a Logical view 183
4.2 Assigning managed systems to a Logical view. 189
4.2.1 Assigning a managed system to a specific level. 190
4.2.2 Using managed system inheritance 192
4.2.3 Adding views with explicit managed system 193
4.3 Assigning situations to a Logical view. 193
4.3.1 Creating and deleting a situation 194
4.3.2 Associating and dissociating a situation 195
4.3.3 Advice in a situation 196
4.4 Defining the links. 201
4.4.1 Introducing the links 201
4.4.2 Defining an absolute link. 203
4.4.3 Defining a relative link. 211
4.4.4 Defining an advanced link. 214
4.5 Style in the Logical view 241
4.5.1 Using the standard Tivoli Enterprise Portal. 241
4.5.2 Using the Smart Bank showcase style 242

4.5.3 Setting up a style in a graphic view 243
4.5.4 The Smart Bank showcase.css style file 245
4.5.5 The example: The Smart Bank showcase Logical view 246
4.6 Logical view: Lessons learned 252

Chapter 5. Scenarios and use cases 257
5.1 Core system transformation (service-oriented architecture) 258
5.1.1 Introducing the architecture through Tivoli Enterprise Portal 258
5.1.2 Service-oriented architecture view 260
5.1.3 Integration hub: WebSphere MQ 268
5.1.4 Process integration 269
5.1.5 Summary 271
5.2 Optimization of IT resources (On/Off Capacity on Demand) 272
5.2.1 The Operations view 273
5.2.2 Monitoring the monitoring environment 276
5.2.3 Monitoring the network 277
5.2.4 IBM System z9-109 utilization 278
5.2.5 Summary 280
5.3 Branch transformation and infrastructure simplification (heterogeneous view) 280
5.3.1 The Branch view 282
5.3.2 The Linux utilization view 283
5.3.3 Summary 284

Related publications 285
IBM Redbooks 285
Other publications 285
Online resources 285
How to get IBM Redbooks 286
Help from IBM 286

Index 287

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products must be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

© Copyright IBM Corp. 2006. All rights reserved.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
BladeCenter®
CICSPlex®
CICS®
Component Business Model™
DB2 Universal Database™
DB2®
Distributed Relational Database Architecture™
DRDA®
DS6000™
DS8000®
ESCON®
eServer™
GDPS®
Geographically Dispersed Parallel Sysplex™
HiperSockets™
HyperSwap™
IBM Component Business Model™
IBM®
MQSeries®
NetView®
OMEGAMON II®
OMEGAMON®
OS/390®
Parallel Sysplex®
POWER4™
POWER®
Rational®
Redbooks®
Redbooks (logo) ®
Resource Link™
System i®
System p®
System x®
System z9®
System z®
Tivoli Enterprise Console®
Tivoli®
TotalStorage®
VTAM®
WebSphere®
z/OS®
z/VM®
z9®
zSeries®

The following terms are trademarks of other companies:

SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Interchange, Red Hat, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

VMware, the VMware "boxes" logo and design are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

EJB, Enterprise JavaBeans, J2EE, Java, JavaBeans, JavaServer, JDBC, JRE, JSP, JVM, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

ESP, Expression, Internet Explorer, Microsoft, PowerPoint, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

All organizations will eventually face the necessity to refresh ageing technology. IBM® infrastructure solutions are a combination of hardware, software, services, and solutions designed to get the most out of the resources and help move from the existing technology to a new technology, without throwing away the existing investments.

IBM infrastructure solutions simplify the IT infrastructure of businesses and enhance business continuity and systems management through the following features:

- Ensuring better utilization

  When faced with an expanding workload, clients may invest in additional hardware although all they may have to do is obtain information about how to get more out of their existing assets.

- Aiding effective management

  Having a wealth of IT resources is one thing, being able to monitor, manage, and allocate virtual resources and physical resources to dynamically address the business requirements is another.

- Promoting rapid deployment

  This is the ability to adapt to the changing competitive landscape in which businesses either succeed or fail. Usually, clients do not know where their next challenge will come from and what kind of tools will be required to address it.

- Ensuring business continuity

  A company's IT infrastructure must ensure that business continues to function well under all conditions and circumstances.

- Enhancing security

  A secure infrastructure must create the necessary conditions for an open, dynamic, and shared environment in which collaborative processing takes place between employees, clients, suppliers, and partners.

- Promoting compliance

  Regulatory frameworks evolve every day, forcing businesses to incorporate changes without affecting business opportunities.

IBM created the Smart Bank showcase to demonstrate to retail banking clients the benefits of IBM products, services, and solutions in addressing their key business issues. The Smart Bank showcase covers different solution areas such as IT service management, service-oriented architecture, optimization of IT resources, information life cycle management, information warehouse, and business continuity. The Smart Bank showcase is an implementation of these solutions in a real situation.

This IBM Redbooks® publication looks at the infrastructure monitoring and infrastructure management provided by IBM Tivoli® Monitoring V6 and IBM Tivoli Composite Application Manager V6. Although the role of these tools is mainly to optimize the IT infrastructure, they play an important role across other solution areas in helping manage heterogeneous infrastructure.

© Copyright IBM Corp. 2006. All rights reserved.

This book includes the following chapters:

- Chapter 1, “Introducing the technical architecture of the Smart Bank showcase” on page 1 describes the architecture of the Smart Bank showcase and discusses the choices, the components, and the IBM Tivoli Monitoring design.
- Chapter 2, “IT environment of the Smart Bank showcase” on page 51 provides details about the components and the implementation of the showcase.
- Chapter 3, “Using Physical views” on page 121 describes the Physical views used in the showcase.
- Chapter 4, “Customizing the Logical views” on page 181 describes the customization of the Logical views used in the showcase.
- Chapter 5, “Scenarios and use cases” on page 257 describes the scenarios presented in the showcase.

The team that wrote this IBM Redbook

This book was produced by a team of specialists from around the world working at the IBM European Products and Solution Support Center (PSSC), Montpellier, France, in collaboration with the IBM International Technical Support Organization (ITSO) Poughkeepsie center. The PSSC is the largest support center in Europe for IBM eServer™, including IBM System z®, IBM System p®, IBM System i®, IBM System x®, and IBM TotalStorage®.

Christian Matthys has spent more than 25 years at IBM as a System Engineer, working with large mainframe-oriented clients in France. He spent three years as an ITSO Project Leader on assignment in Poughkeepsie, NY, USA. In 2000, he joined the Europe, Middle East, and Africa (EMEA) Design Center for On Demand Business at the PSSC in Montpellier, France, working on clients’ projects to make use of leading edge technologies. He works as a Project Leader for the ITSO Poughkeepsie Center, NY, USA. He is based at the PSSC. He is a certified Consulting IBM IT Specialist.

Guillaume Arnould joined IBM in 1996 and started working in the IBM System z Manufacturing Test Engineering Department before spending two years in an international assignment in Poughkeepsie, NY, USA. In 2001, he joined the IBM PSSC team in Montpellier, France, to work as a Performance Expert on IBM DB2® for z/OS® client benchmarks, where he has been since then. He is a DB2 for z/OS IT Specialist in the infrastructure solutions department at IBM PSSC. Guillaume is a Technical Team Leader in the Smart Bank showcase project. He is responsible for leading the new developments and production platforms. He holds an engineering degree in automation, electronics, and computing from Ecole Superieure en Informatique, Electronique et Automatisme (ESIEA), Paris. His areas of expertise include DB2, DB2 data sharing, performance tuning, and administration.

Neil Ashworth is a Senior Certified IT Architect working in the IBM Design Center for On Demand Business at IBM PSSC, Montpellier, France. He has been with IBM since 2001. He has over 10 years experience in the financial services sector, working on international banking client projects. Neil specializes in integration technologies and techniques around the IBM System z platform.

Stefano Delle Chiaie is a certified IT Specialist based in IBM PSSC, Montpellier, France. He has 18 years of IT experience, half of which is in the aerospace industry as a client. He joined IBM in 1995 and has since performed various benchmark and performance studies for clients in the financial sector. He currently works in the IBM PSSC’s industry infrastructure solutions group.

Eric Devezeaud is an Advisory IT Specialist, working in IBM Information Technology Services in Geneva, Switzerland. He joined IBM in 1999 and has more than 15 years of experience in the IT sector. He holds a Masters degree in Computer Science from the University of Savoie, France. He specializes mainly in the performance, availability, and configuration management domains. He is a Project Leader for designing and implementing solutions based on Tivoli products such as IBM Tivoli Monitoring, IBM Tivoli Enterprise Console®, IBM Tivoli NetView®, and IBM Tivoli Configuration Manager.

Fabrice Jarassat is an IBM Customer Information Control System (CICS®) and IBM CICSPlex® IT Specialist within the infrastructure solutions department at the PSSC in Montpellier, France. He joined IBM in 2000. Before joining IBM, he worked at Mag-Info, which is a part of the French Galeries Lafayette group, responsible for managing the transaction system based on CICS. He holds a degree in computing from Ecole Superieure Informatique Professionnel, Paris. His areas of expertise include CICS, IBM CICSPlex, CICS e-business technologies for z/OS, and CICS Tools. He is a Subject Matter Expert in CICS Web enablement.

Francois Lebe is an IBM System z Technical Pre-sales Specialist, working at the IBM office in Paris. He has been working on Candle since 1999. He joined IBM in 2004 when Candle Corporation was acquired by IBM. He has over 20 years of experience in the software business. Francois specializes in supervision and automation of System z platforms. He holds a degree from the Institute Universitaire De Technologie, specializing in IT. He has written extensively for French clients on how to supervise System z platforms using the IBM Candle automation and IBM OMEGAMON® products.

Detlef Wolf is an IT Specialist for IBM Tivoli System Automation, working for the Tivoli Technical Sales team based in Munich, Germany. He has 16 years of experience in the IT industry. Starting in an application developing department in the medical care industry, he became a System Programmer for z/OS in an insurance company. In 2000 he joined Candle and focused on IBM WebSphere® products, and the management of these systems. Since 2004 he has been a part of the IBM team in Germany, working together with major financial services clients on deploying IBM Tivoli products in their environment. He holds a Graduate Engineer Degree in Computer Science from the University of Erlangen-Nuernberg.

Thanks to the following people for their contributions to this project:

Fred Winegust
IBM Canada

James Goethals
IBM USA

Luis Aused López
IBM Spain

Nathalie Defez
IBM France

Peter Fowler
IBM USA

Robert Honer
IBM USA

Sharmela Pattabiraman
Editor, ITSO, Bangalore, India

Become a published author

Join us for a two-week to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will team with IBM technical professionals, Business Partners, and/or clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

To find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our IBM Redbooks to be as helpful as possible. Send us your comments about this IBM Redbook or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review IBM Redbook form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ; HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Introducing the technical architecture of the Smart Bank showcase

This chapter describes the key business drivers for the Smart Bank showcase, the features we wanted to showcase, and how we used the IBM Tivoli infrastructure and application monitoring for delivering the solution.

This chapter also discusses the showcase proof points in the context of IBM infrastructure and business solutions. As part of this discussion, this chapter discusses the key architectural decisions and components involved in the Smart Bank showcase because these form the basis of the elements to be monitored.

© Copyright IBM Corp. 2006. All rights reserved.

1.1 Understanding the Smart Bank showcase

The official name of the project is Building a Smart Bank Operating Environment. For ease of use, we substitute this with Smart Bank showcase in this book.

The title of this IBM Redbook starts with the phrase “infrastructure solutions”. In fact, the Smart Bank showcase was initiated and designed before IBM started the infrastructure solutions initiative. However, Smart Bank showcase not only fits well with the infrastructure solutions strategy, but also crosses into the IBM business solutions strategy. This book positions the Smart Bank showcase within these IBM initiatives and architectures.

1.1.1 Overview of IBM infrastructure solutions

IBM offers a portfolio of solutions that focus on optimizing your infrastructure. These solutions are specifically tailored for the financial services sector. With these solutions, IBM helps financial institutions achieve the following goals:

- Enhance business resilience and security to protect and strengthen your institution
- Increase business flexibility to drive growth through new capabilities and enhanced value propositions
- Optimize IT environments to enhance efficiency across your company’s IT resources, including voice, data, and network

Within these three subject areas, IBM offers eight solutions that incorporate software, hardware, and services. These are shown in Table 1-1.

Table 1-1 Infrastructure solutions

Infrastructure solution	Brief description
IT service management	Reduces IT cost by automating, integrating, and optimizing change processes
Service-oriented architecture (SOA) infrastructure	Increases flexibility and reduces complexity by integrating applications across disparate systems
Information warehouse	Delivers complete, consistent, trusted, and actionable information
User platform	Provides secure, personalized access anytime, anywhere
Optimization of IT resources	Creates a responsive IT infrastructure that is easier and less expensive to manage, upgrade, and run
Information life cycle management	Manages information from creation to disposal on the most efficient and cost-effective IT infrastructure
Business continuity	Assures that critical business processes, IT systems, and networks are continuously available
IT security	Understands, mitigates, and manages security risks

The rest of this book describes how Smart Bank showcase covers a number of different solution areas. However, the main purpose of this book is to look at infrastructure monitoring and management provided by IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6. These tools help optimize your IT infrastructure and are also useful across other solution areas.

Figure 1-1 maps the IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6 to the infrastructure solutions.

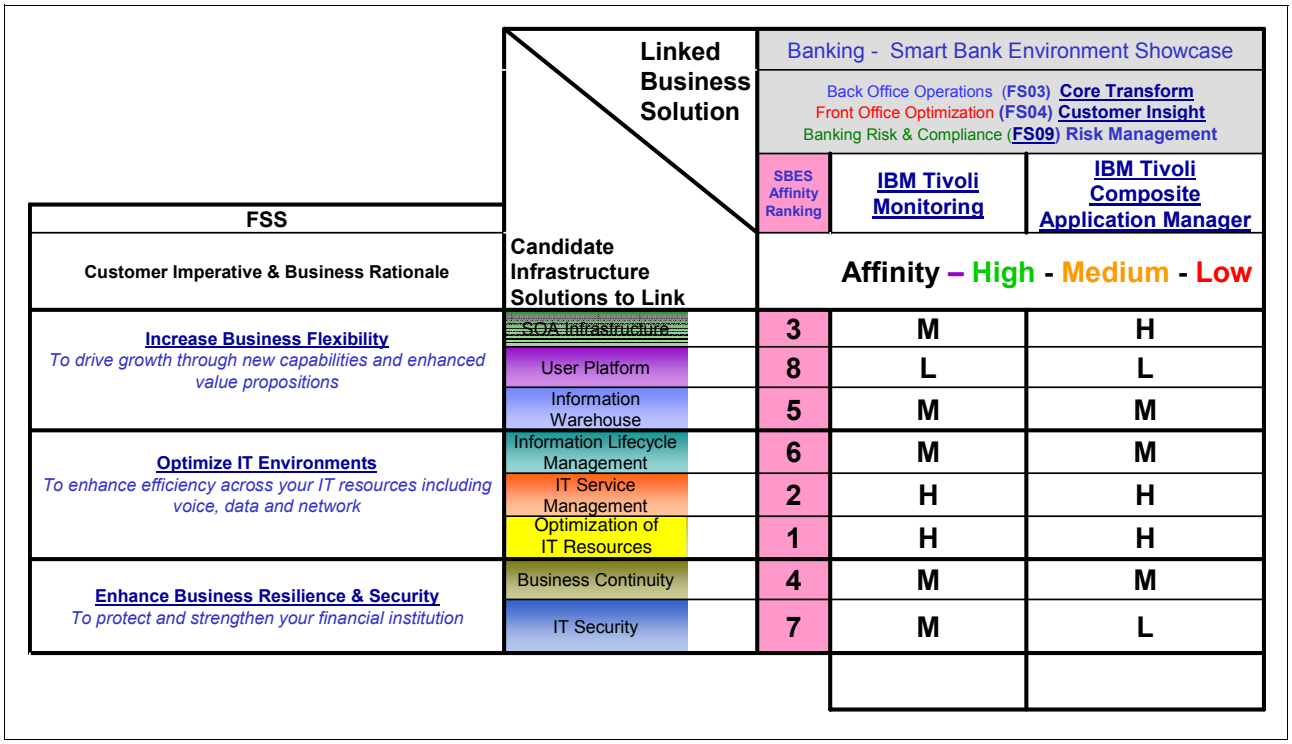

Linked Business Solution
Banking - Smart Bank Environment Showcase
Back Office Operations (FS03) Core Transform
Front Office Optimization (FS04) Customer Insight
Banking Risk & Compliance (FS09) Risk Management
FSS
Customer Imperative & Business Rationale
Candidate Infrastructure Solutions to Link
SBES Affinity Ranking
IBM Tivoli Monitoring
IBM Tivoli Composite Application Manager
Affinity – High - Medium - Low
Increase Business Flexibility
To drive growth through new capabilities and enhanced value propositions
SOA Infrastructure
3 M H
User Platform
8 L L
Information Warehouse
5 M M
Optimize IT Environments
To enhance efficiency across your IT resources including voice, data and network
Information Lifecycle Management
6 M M
IT Service Management
2 H H
Optimization of IT Resources
1 H H
Enhance Business Resilience & Security
To protect and strengthen your financial institution
Business Continuity
4 M M
IT Security
7 M L

Figure 1-1 Affinity mapping of IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6

Systems management

Within the IBM infrastructure solutions reference architecture, one of the key architectural decisions around optimizing IT environments is systems management, which aims to perform the following functions:

- Align the IT infrastructure with business goals
- Increase resource utilization
- Improve personnel productivity
- Increase application availability
- Create an infrastructure that is driven by business priorities
- Create a flexible infrastructure that adapts to changing business requirements, and gain more control of that infrastructure to provide more business agility

Before going into the details of systems management and monitoring, let us consider what we are monitoring and what we are trying to show.

1.1.2 Objective of the Smart Bank showcase

The Smart Bank showcase was created at the IBM European Products and Solution Support Center (PSSC) in Montpellier, France, as a vehicle to show the retail banking clients the value of IBM infrastructure solutions in addressing their key business issues. As such, it was a project whose scope and target audience were spread across the world, directed by the IBM Financial Services Sector (FSS) and the IBM Systems Technology Group (STG).

Following were the objectives of the project:

- To demonstrate the value of IBM infrastructure solutions in the context of retail banking
- To provide a tool to engage the clients and interest them in more projects
- To focus on the business value and develop proof points to show how the infrastructure solutions can help deliver key banking projects
- To demonstrate the proof points at operational volumes typical of a European bank, which includes the following:
  - Six million clients
  - 12 million accounts
  - Average daily online transaction throughput of about 300 transactions per second, rising to a peak of up to 800 transactions per second
  - Constant background batch workload setup as the lowest priority workload during the demonstration
- To provide a platform for internal IBM collateral, testing of concepts, and documentation. In this context, the following points are relevant:
  - We participated in the z9®-109 Early Support Programme (ESP™) project, testing the new machine with near operational volumes and utilization
  - We provided an analytical database developed in Montpellier based on the Information FrameWork (IFW) Banking Data Warehouse (BDW) to other IBM teams
  - This book is the first piece of external collateral to be created from the project

We were able to leverage the resulting database from an Independent Solution Vendor (ISV) (Fidelity) benchmark conducted in Montpellier. This, plus the agreement with the ISV to use their application software to represent the core system in our retail bank, launched the project.

As the project grew to cover new business proof points, we engaged other ISVs to provide the application functionality to highlight our infrastructure.

The following is a list of the ISVs and IBM assets engaged in the Smart Bank showcase project:

- Fidelity Corebank V4.2, a real-time retail banking application based on a physical implementation of the Financial Services Data Model (FSDM) that forms the foundation of the Information FrameWork (IFW) model from IBM Software Group (SWG) in Dublin.
- Fair Isaac TRIAD V8.0, a risk calculation engine
- Siebel® Business Analytics V7.7.1, a business intelligence application
- Stonesoft Stonegate V2.2.7, a Linux® firewall security software
- IBM Banking Data Warehouse (BDW) from IFW, built on the FSDM data model
- IBM PSSC, Montpellier, resources, experts, hardware, and software

1.1.3 The banking sector's current key business drivers

This strategic input came from the Financial Services Sector (FSS) and the Global Business Services (GBS) and its IBM Institute of Business Value.

The CEOs of retail banks currently face major issues and challenges, the most important of which are as follows:

- Banks are seeking new revenue growth and client profitability strategies to compete in a competitive environment
- Outdated core systems and poor merger and acquisition (M&A) integration are making it difficult for banks to respond to the changing conditions
- New threats and regulations are necessitating investment in compliance measures and new risk management strategies

Table 1-2, Table 1-3, and Table 1-4 look at these issues in detail, identifying some of the challenges and the key recommendations from IBM GBS to banks.

Table 1-2 Revenue growth and client profitability

Industry issue:	**1) Revenue growth and client profitability**
Challenge for banks	► To be able to exploit client data so that they can increase the profitability of existing clients through targeted product and service offerings ► Identify untapped revenue sources and be able to develop unique value propositions based on their business intelligence, and thereby increase competitive differentiation
Recommendations	► Shift from product focus to client needs ► Client segmentation/profitability focus ► Multichannel delivery service ► Innovative and flexible product packages and solutions using core competencies

Table 1-3 Outdated core systems and poor M&A integration

Industry issue:	**2) Outdated core systems and poor M&A integration**
Challenge for banks	► Adapt the core systems and integrate disparate platforms and processes to reduce costs and support changing demands ► Reduce the high fixed costs of noncore processes while increasing efficiency and flexibility
Recommendations	► Reduce costs and improve efficiency by leveraging existing mission critical core systems with better integration ► Replacing in a modular fashion certain components of outdated systems ► Outsource noncore IT infrastructure ► Share best-in-class capabilities across product silos - horizontal integration

Table 1-4 Investment in compliance and risk management

Industry issue:	**3) Investment in compliance and risk management**
Challenge for banks	► Ensure compliance with pending regulations while enhancing existing systems and processes ► Develop business resiliency and operational high-risk mitigation capabilities with minimum expenditures

Industry issue:	3) Investment in compliance and risk management
Recommendations	► Build operational efficiency objectives into risk management strategies. Operational efficiency becomes a key performance metric for risk management. ► Share cost of disaster recovery/business continuity and regulatory compliance capabilities. Regulatory compliance a matter of survival. So banks must pool their resources to provide resilience capabilities. ► Create a risk framework where risk management investments (databases, analytical engines) can be leveraged by client relationship management and business intelligence.

With these key trends in the background, we saw certain emerging and declining trends over shorter timeframes. In 2005, for example, transforming the bank branch was a key issue across the financial services sector and not just retail banking, because financial institutions saw the bank branch as a key sales and client advice and relationship outlet.

Within the financial services sector, we recognized banking, financial markets, and insurance. However, behind the trend referred to as “rolling out the red carpet to clients in branches”, key trends such as increasing client profitability, multichannel delivery, and so on, were also identified.

With these issues in mind, we realized that the key drivers for spending by banks are aligned across the same areas, as shown in Figure 1-2.

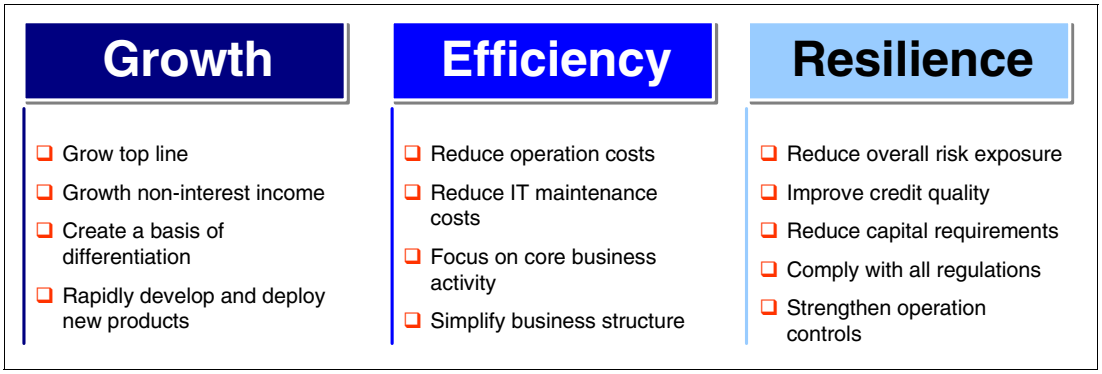


Figure 1-2 Current drivers of spending by banks

1.1.4 The banking sector's future key business drivers

A research conducted by the IBM Institute of Business Value among CEOs of leading institutions indicates significant challenges at different levels in the retail banking industry as we look ahead to the year 2015. The challenges include the following:

- At the client level
  - Clients will become increasingly individualistic. For the financial institution, the capability to make intelligent client segmentation becomes a key capability as it becomes more difficult for banks to pin-point broad groups of clients. This business intelligence allows institutions to target their products more carefully to specific segments.
  - Clients will also become more fickle with regard to brand loyalty, when their ability to access more information about different competitive products across the market increases. They will increasingly control their relationships with financial institutions, making the rapid creation of products aligned to the requirements of client groups another key ability of the financial institution.
- At the financial institution level
  - Financial institutions will become increasingly consolidated, with large international banks continuing to acquire mid-tier banks, and mid-tier banks merging to become more competitive, often seeking foreign investors.
  - The banking landscape will change because it will become increasingly hard to remain competitive in the middle ground, giving rise to universal banks, industry specialists focusing on specific product areas, community banks focusing on client relationships, and the increasing importance of nonbanking entrants leveraging their existing client relationships and distribution networks, including retailers and car manufacturers.

These trends will drive growth strategies that are focused on innovation with regard to the following points:

- Product or service innovation
- Client intimacy and knowledge of client groups
- New markets and channels
- Diversification strategies

Figure 1-3 illustrates these points.

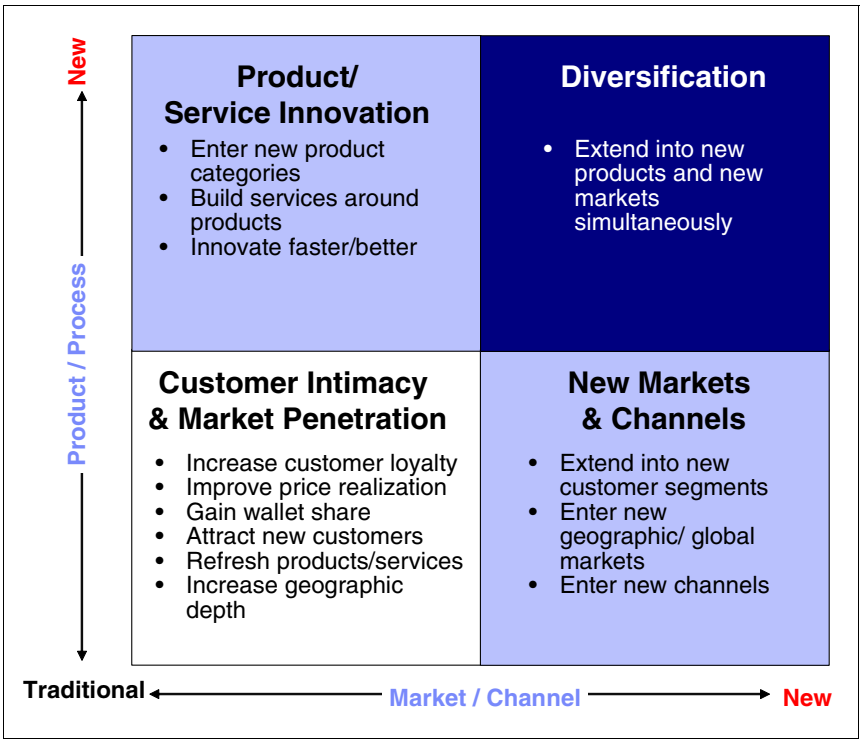


Figure 1-3 Strategies for innovative growth

The future trends forecast by IBM GBS builds on the current trends and strategies that are seen in the marketplace. Both the future and the current trends provide the direction for the Smart Bank proof points.

1.1.5 The way the IBM Financial Services Sector addresses these issues

At the inception of the project, we wanted to take clients through a set of business scenarios they would recognize and realize as being strategic for their business. We called these *proof points* because we wanted to demonstrate how IBM architectural concepts, products, and techniques address these business problems and provide business value to the client. We use the business trends and issues described earlier to reinforce the message of what we are providing and why we are providing the same.

The Financial Services Sector has developed a comprehensive architecture and framework to address these issues. Although this project is clearly focused on infrastructure, the application layer must also be able to deliver on many of the key business proof points.

A distinction is made between business solutions and infrastructure solutions, which can be mapped broadly against the service offerings from GBS for the business solutions and the IBM Global Technology Services (GTS) for the infrastructure solutions. However, when you look at the business solutions for banking within Financial Services Sector (FSS), the borders are blurred, as shown in Figure 1-4.

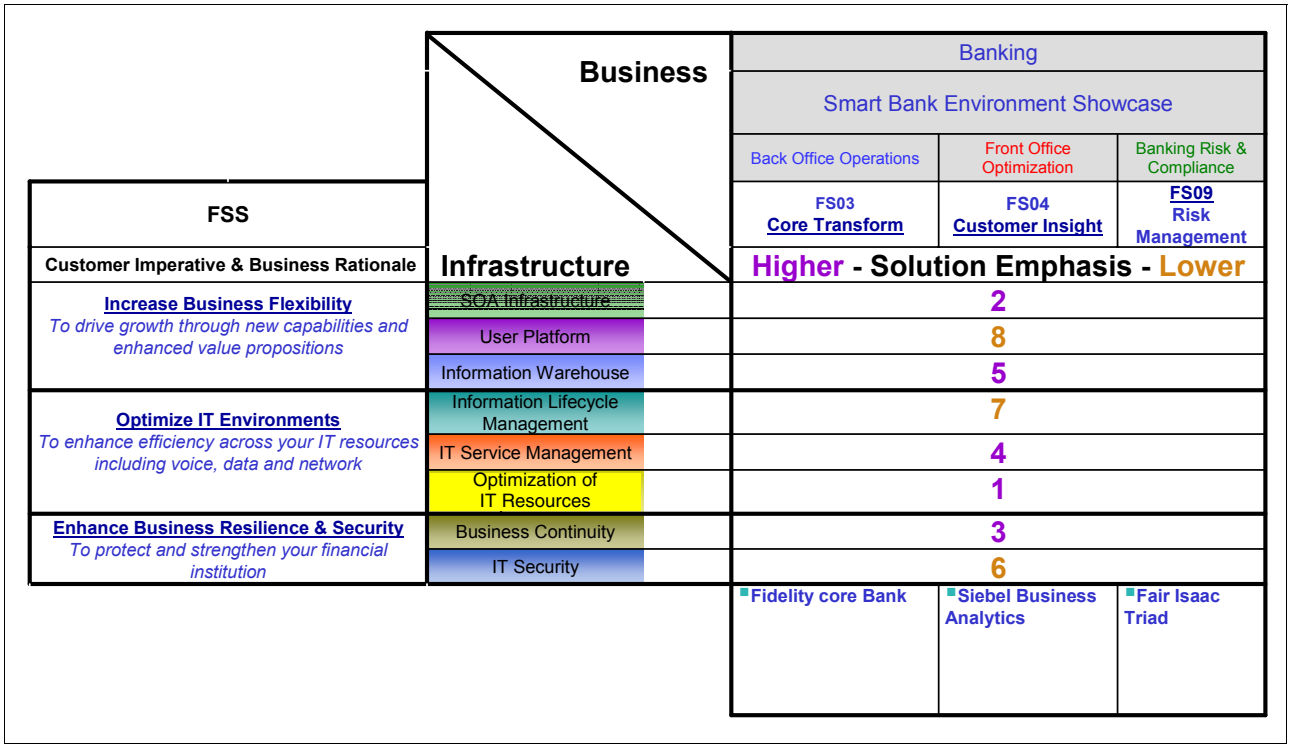


Figure 1-4 FSS business solutions' affinity with the infrastructure solutions mapped to the Smart Bank

Gray areas, where business and infrastructure solutions merge, exist, including the following:

- Addressing risk and compliance with analysis of data (The Information Warehouse Infrastructure Solution is required.)
- Addressing core systems transformation, service-oriented architecture (SOA) (Infrastructure and IT Service Management is used.)

The solutions, whether business or infrastructure, remain practical initiatives that IBM uses to help its clients address the key business issues outlined.

1.1.6 Demonstration proof points: A day in the life of a bank

For this demonstration, we took the important strategies and tried to tackle them in a logical order even as we created our environment, trying to address a proof point for retail banking each time.

The following proof points are addressed in the Smart Bank demonstration:

- Multichannel transformation

  The first thing we did was to integrate our modern core retail banking system that was provided by Fidelity to our channels in order to allow our clients to transact with us. Multichannel transformation involves the following:

  - The ability to reuse business logic across more than one delivery channel and open new delivery channels or enterprise product offerings rapidly
  - The ability to manage these channels centrally and collect metrics in real time. This may be used for billing lines-of-business based on usage
  - To control the quality of service and user experience across these channels
  - To provide a resilient, secure, scalable, and highly available business service for these delivery channels

  This step involves front office optimization from the business perspective and the SOA infrastructure and user platform from the infrastructure perspective.

- Branch transformation

  This implies multiple tasks, including the following:

  - For the branch channel, we adopted a centralized model, where the branch servers are physically located on centralized servers in order to help simplify the environment and reduce costs.
  - This proof point focuses on horizontally scalable applications deployed in a distributed environment, perhaps in a regional hub, or in a central server environment.
  - The branch transformation proof point also inherits the points from the multichannel transformation point and shows end-to-end integration across heterogeneous platforms.

  This step involves front office optimization from the business perspective and SOA infrastructure and user platform optimization from the infrastructure perspective.

- Rapid product development (quicker time-to-market with SOA)

  After achieving the flexibility accorded by the multichannel architecture, we tackled the new programming model and the SOA. The key driver for this proof point was rapid development (quick time-to-market) of the new business products.

  This development implies the following:

  - Leverage the core system functionality by exposing it as a service that is to be used by a business process
  - Build new business processes based on the existing core system functionality and the new functionality created in the new programming model (Java™ 2 Platform, Enterprise Edition [J2EE™]). Rapid product development leads to a discussion around core systems transformation.

  This step involves SOA infrastructure and IT service management from the infrastructure perspective.

- Core systems transformation

  This includes the following:

  - Identifying the core competencies and functions encapsulated into the core systems and then leveraging them through an SOA
  - Identifying back office areas that can be optimized for more efficient functioning. With access to mission-critical core system functionality through SOA and using the new programming model of J2EE and open standards and processes, it is possible to optimize these functions and add value, such as:
    - Greater integration for cross-selling
    - Enabling transformation of the core operation
    - Providing system flexibility to deal in a better way with regard to new products and regulatory compliance

  This step involves core systems transformation from the business perspective and SOA infrastructure and IT service management from the infrastructure perspective.

- Client insight (business intelligence)

  This includes the following:

  - Creating client segmentation using an analytical banking data warehouse. This includes identifying key client groups to whom a new product is offered.
  - Creating an architecture to update the analytical database in near real time in order to help businesses take more informed decisions based on up-to-date information

  This step involves front office optimization from the business perspective and information warehouse and information life cycle management from the infrastructure perspective.

- Regulatory compliance

  This includes the following:

  - Regulatory reporting from the risk framework created by leveraging the analytical database created for client insight
  - Showing how an enterprise can meet some of the demands of Basel II through the use of a central banking data warehouse (analytical database)

  This step involves banking risk and compliance from the business perspective and information warehouse and business continuity from the infrastructure perspective.

- Business continuity (operation risk and business resiliency)

  This includes the following:

  - Showing how an enterprise can react to unplanned outages, even as it maintains data integrity, consistency, and security
  - Creating a highly available environment with hardening of platforms, redundancy, and recovery
  - Allowing scalability, both horizontal and vertical, and planning for prioritizing workloads and systems

  This step involves banking risk and compliance from the business perspective and business continuity from the infrastructure perspective.

- Infrastructure simplification

  This includes the following:

  - Infrastructure simplification is an underlying message behind the entire demonstration of the monitoring and management of a system, which can use virtualization and autonomic technologies
  - Infrastructure simplification, along with systems management, is the proof point that this IBM Redbook helps address. It also provides the mechanism through which the other proof points are demonstrated.

  This step involves optimization of IT resources and information life cycle management from the infrastructure perspective.

This book does not claim to deliver the ultimate solution to all these problems. It shows how IBM strategies and infrastructure can be used to address their problems with the appropriate use of technologies and a set of example ISVs performing certain business functions.

A day at our bank

This book uses a storyboard to drive the demonstration and to tackle the proof points clearly. This story involves a day's activity in the bank. It starts at 06:00 hrs in the morning[1]. The bank's clients are either returning home from the night shift, leaving the Spanish nightclubs, or are starting the day. Table 1-5 shows details of the events that we typically run through for the Smart Bank showcase.

Table 1-5 Story board relating to a day in the life of the Smart Bank

Time line[a]	Event	Description
06:00 a.m.	Setting the scene: Early morning at the bank	We show low channel traffic coming from the automated teller machines (ATMs) and Internet channels, with clients withdrawing cash for the day or accessing the Internet bank. (We show the injection mechanism, perform a business transaction or two from our Internet bank, and emphasize that this is a real workload on a live operational system.)
09:00 a.m.	Branches open: Workload increases	A new channel comes online and the clients start transacting through the branches. (We introduce our enterprise management - IBM Tivoli Enterprise Portal (TEP) - views and show the branch channel arriving from a workload point of view and highlight the quality of service (QoS), which can be measured across the client-facing channels.)
10:00 a.m.	Business intelligence: Client segmentation	The bank's analysts have, with the help of business intelligence software, detected a trend in the mortgage portfolio, and continue to identify client segments and promotional opportunities. (Now we launch into client profitability, revenue growth, and the client segmentation proof point, using Siebel Business Analytics.)

[1] We recognize that many banks operate across different time zones and that this can even out or blur the peaks and troughs of a daily workload. For reasons of simplicity and for a more dramatic dynamic demonstration, we restricted the demonstration to one time zone for now.

Time line[a]	Event	Description
10:30 a.m.	Near real-time business analytics data	How have we created the architecture for our analytical database and achieved a near real-time update of our analytical database? (We show near real-time analytics update speed through simple Structured Query Language (SQL) queries from Siebel Business Analytics to the banking data warehouse, and describe the architecture.)
10:45 a.m.	Leverage the new programming model	The bank wants to rapidly create new business products to offer its clients. We leverage the capabilities presented with SOA to do this, along with leveraging the existing mission-critical applications. (We show the SOA workload running and describe the architectural components and techniques.)
11:00 a.m.	Introduce business performance dashboards	A new capability made possible by SOA, a dashboard for business performance, helps measure the performance of the channels. (We show the business dashboard with near real-time business metrics and describe how we achieve rapid application build.)
12:00 p.m.	Manage lunch time volumes	Lunch time workload increases. (We show how workload management can automatically maintain QoS and service-level agreements based on business priorities.)
13:00 p.m.	New promotion launch: Promotion peaks internet traffic	Our promotion is working better than expected. On/Off Capacity on Demand. (We move to the Hardware Management Console to initiate an On/Off Capacity on Demand request. We show the effect of this request on the workload through TEP, and highlight the virtualization capabilities of the platform.)
14:00 p.m.	Business continuity	Some problems arise at this juncture. We discuss the concepts of business continuity, and from an IT point of view, we achieve recovery, redundancy, and hardening. (Using TEP, we run the different scenarios of planned and unplanned outages.)
15:00 p.m.	Credit (Basel II) and operational risk	Having discussed some of the ways in which to counter operational risk, we now turn to credit risk reporting, another aspect of Basel II compliance. (We return to Siebel Business Analytics to reuse the business intelligence framework for regulatory compliance reporting with credit risk.)
17:00 p.m.	Branches close: Transforming a branch infrastructure	For branch consolidation and the ability to have platform independence, we choose a platform based on the quality of service. Simplification of the infrastructure is achieved. (We discuss the concepts of branch transformation and how we have simplified the environment and saved costs.)

a. The time is sometimes contrived for purposes of demonstration. The business event does not have to be restricted only to this timing.

1.2 Architectural choices

This section contains a synopsis of some of the key decisions and thinking behind the architecture we adopted to address our proof points.

1.2.1 Banking clients' current architecture

What architectures do your clients have? If you can build the same type of architecture, you can better simulate the problems and issues that they are trying to solve. Luckily, banks have similar architectures. They are highly complex, but have some key differentiators and similarities that you can use to base your system on. Table 1-6 describes a typical banking architecture.

Table 1-6 Summary of a typical banking architecture

Banking architecture	Description
Centralized model	Most retail banks have a number of branches or outlets that communicate back to a central or regional headquarters. The IT environment follows a similar model, with the outlets/channels being termed as front office, and central processing and settlement being termed as back office.
Many channel types	Many technically and physically different methods of distributing products and services to the clients
Large centralized databases	Often, several data stores holding client, account, product, financial, and analytical information
Multiple interfaces for applications, suppliers, partners, and recent acquisitions	Relationships with numerous external organizations from multiple internal specialized offices, results in various technical interfaces
Years of investment in mission-critical core systems	Even if they buy solutions from ISVs, many banks spend years trying to enrich the functional capabilities of their core systems
Various IT platforms	Often, many different hardware platforms and operating systems
Heavy input/output-intensive workloads	One characteristic of the banking workload is the requirement to balance the books after a day's activity, which results in a batch window in which all the I/O-intensive workload must be completed for the day in order to feed the financial settlement systems.

1.2.2 Reference architectures and models

We had a number of IBM reference architectures and models at our disposal, including the following:

- On demand infrastructure solutions reference architecture
- IBM Service-Oriented Architecture (SOA)
- IBM Financial Services Architecture (FSA)
- IBM Component Business Model™ (CBM) view of retail banking
- IBM Information FrameWork, Critical Business Processes, Banking Data Warehouse, and Financial Services Data Model

1.2.3 Functional requirements

The high-level functional requirements we looked for included those functions that support the proof points and fall under the category of business solutions. Table 1-7 describes the most common functional requirements.

Table 1-7 High-level functional requirements

Functional requirement	Description
Core retail banking application	A fully implemented and functional core retail banking system to simulate the sort of applications that most retail banks have running at the heart of their operation. These are the operational account management systems that provide the retail products to be used by the channels, and record client involvement across the banks' portfolio in current accounts, savings accounts, deposit accounts, loans, and other credit facilities for the general public and for specific client segments.
Service layer, reusable application programming interfaces (APIs)	An application layer that allows the reuse of the core system functionality in a multichannel, and later in an SOA
Different channels	The ability to inject workload across a number of different channels to our core system in order to represent a subset of the many channels most retail banks offer their clients and partners
Rapid application build	The capability to build an application component to demonstrate the power of the new SOA
Credit risk calculation	An application to calculate credit risk and other metrics used in calculating a bank's exposure to its clients
Creation of an analytical database	Selecting a database model and creating a central analytical database that can be used for risk analysis and business intelligence
Interface to credit risk calculation	Populating the analytical database with credit risk details and feeding the analysis from the analytical database
Business intelligence application	Selecting an application to provide a business intelligence solution and analytics that can access a database through open standards and not prescribe any proprietary database
Business performance dashboard	Creating a simple dashboard for business metrics using open standards and leveraging the SOA
Basel II credit risk reporting	Selecting a reporting tool to create the Basel II reports without the necessity for any proprietary database to access the analytical database of choice
Batch workload	A simulation of a typical I/O-intensive batch workload against the same operational core system database that characterizes the banking environment
Reduce time-to-market of solutions	Leveraging information insight, development tools, and an SOA. Reusing assets where possible.

1.2.4 Nonfunctional requirements

What qualities of service do we expect for our infrastructure and solution? Table 1-8 describes the most common nonfunctional requirements.

Table 1-8 Nonfunctional requirements

Nonfunctional requirement	Description
Provide systems management and monitoring for the enterprise	Creating a Logical view of the infrastructure to monitor and managing the infrastructure in as automated a manner as possible
Workload management-to-business goals	Maintaining the quality of service across the channels and critical business processes to ensure client satisfaction and rapid response-to-business goals
Near real-time update	Using the infrastructure to create a near real-time update of the analytical database to provide more up-to-date information for business analysts and risk reporting
Straight-through processing	Real-time processing of and response to online channel transactions and operations
Business continuity	Providing for a highly available and resilient platform and being able to demonstrate this
Security	Providing a secure infrastructure
Transactionality	Ensuring that data integrity is maintained within the transactional unit of work, and ensuring that they have atomicity, consistency, isolation, and durability (ACID) properties
Scalability	The architecture must be able to scale in as near linear fashion as possible so that the cost of per unit of work remains constant even if the workload increases
Language and platform neutrality	Maximizing the independence of applications from the underlying platform so that the appropriate platform is chosen, based on the qualities of service required
Virtualization	Benefitting from virtualization technologies where possible, in order to simplify the environment and increase the efficiency and flexibility with which the environment is managed and provisioned

1.2.5 Key architectural decisions

Note that the ISVs were chosen by the FSS based on the fact that their applications and technologies fit our architectural requirements. However, they are not the only ISVs that could have played the role.

Table 1-9 lists some of the key architectural decisions that were taken during the creation of the showcase, and which contributed to our current architecture.

Table 1-9 Key architectural decisions

Architectural decision	Description
Central platform	IBM System z9®-109. Adopting the same “mainframe” platform as our numerous clients for the mission-critical core system workload

Architectural decision	Description
Use of open standards to aid integration and platform independence	Using open standards where possible to assist the flexibility and interoperability of the interfaces and the independence of the platforms
Use of industry models	IBM Information FrameWork, Critical Business Processes, banking data warehouse, and Financial Services Data Model
Core retail banking application	Fidelity Corebank V4.2, a modern, real-time core retail banking system based on our industry model and proven to adhere to many of our nonfunctional requirements. This choice was assisted by the availability of the database. Prescribed the Customer Information Control System (CICS) as our transactional subsystem running on z/OS, instead of the Information Management System (IMS).
Business intelligence	Siebel Business Analytics V7.7.1 is a good candidate for an analytics engine. This was run on an UNIX® platform and implemented on IBM WebSphere on IBM AIX® on a POWER4™ blade. It has Open Database Connectivity (ODBC) access to our analytical database.
Credit risk	Fair Isaac TRIAD V8 was a good candidate for our credit risk calculation engine and ran on z/OS, performing the calculations in periodic batch mode
Security firewall	Stonegate Stonesoft in a Linux firewall that we deployed on a standalone Linux logical partition (LPAR) and on Linux machines under IBM z/VM®
Transactional subsystem	CICS and WebSphere Application server on z/OS, with WebSphere ND on Linux
Database	IBM DB2 on z/OS for both our operational databases based on the Financial Services Data Model for Fidelity Corebank and based on banking data warehouse for our analytical database
Central database architecture	Using a central database architecture in order to benefit from data sharing with the cluster (Parallel Sysplex®), the speed of access and data propagation, and the other key qualities of the IBM System z platform, security and high availability.
SOA	Facilitating the flexible integration of new and existing products, contributing to a reduction in time-to-market
Systems management	IBM Tivoli Monitoring V6, because it allows monitoring and managing of all the environments and subsystems, and creates Logical views of this information for an end-to-end Enterprise view

1.3 The architectural components

What is the architecture of the Smart Bank showcase and how are the components of this architecture connected together? This section briefly describes the environment to be monitored and managed by IBM Tivoli Monitoring.

In our case, we chose three key use cases (proof points) to use in the showcase with the newly created Physical and Logical views. These views cover a wide cross-section of IBM Tivoli Monitoring functionality and represent new views for the showcase. They are used to go into more details with regard to specific proof points.

1.3.1 The current architecture of the Smart Bank showcase

The current architecture of the Smart Bank showcase comprises different features, which are described in the following sections.

Core system

Our bank's core system is Fidelity Corebank V4.2, which is a modern retail banking application based on real time update principles to a database based on the Financial Services Data Model from Information FrameWork and physically implemented in DB2 for z/OS. It natively runs on CICS and is predominantly written in Common Business Oriented Language (COBOL). It is supplied with a comprehensive set of application programming interface (APIs) that allow banks to interface with its business functionality. In V4.2, Corebank natively employes a proprietary interface technique using WebSphere MQ as the transport protocol. Figure 1-5 displays the architecture overview diagram from a physical perspective, showing where the core system is implemented in relation to the other systems.

Multichannel architecture

Our multichannel architecture uses the Corebank proprietary WebSphere MQ mechanism to interface to Corebank. We created a Java application to receive our channel requests and interface them to our core retail banking system. This application uses the Struts framework and writes through Java Message Service (JMS) and the Corebank interface Java classes supplied to WebSphere MQ. This application is deployed to the WebSphere Application Server on z/OS for our ATM and Internet channels, and to WebSphere Application Server on Linux (z/VM and IBM BladeCenter®) for our branches. The key architectural message here is the exact reuse of the same core system APIs for all our channels, and the ability to monitor and manage the service-level agreements and workload management across these channels at an enterprise-level.

Core system transformation and service-oriented architecture

Our architecture is then developed using some of the key standards and techniques recommended by IBM for establishing an SOA. Fundamentally, we automatically create services using tools that leverage the core system functionality previously exposed through proprietary APIs and establish a process integration platform through the use of WebSphere Business Integration Server Foundation (WBISF). This uses the Web Services Definition

Language (WSDL) definitions of our services along with Business Process Execution Language (BPEL) in order to link different business functions together, thereby providing a mechanism to transform the use of the core system. Figure 1-5 illustrates the Smart Bank showcase architecture with emphasis on the channels that are discussed.

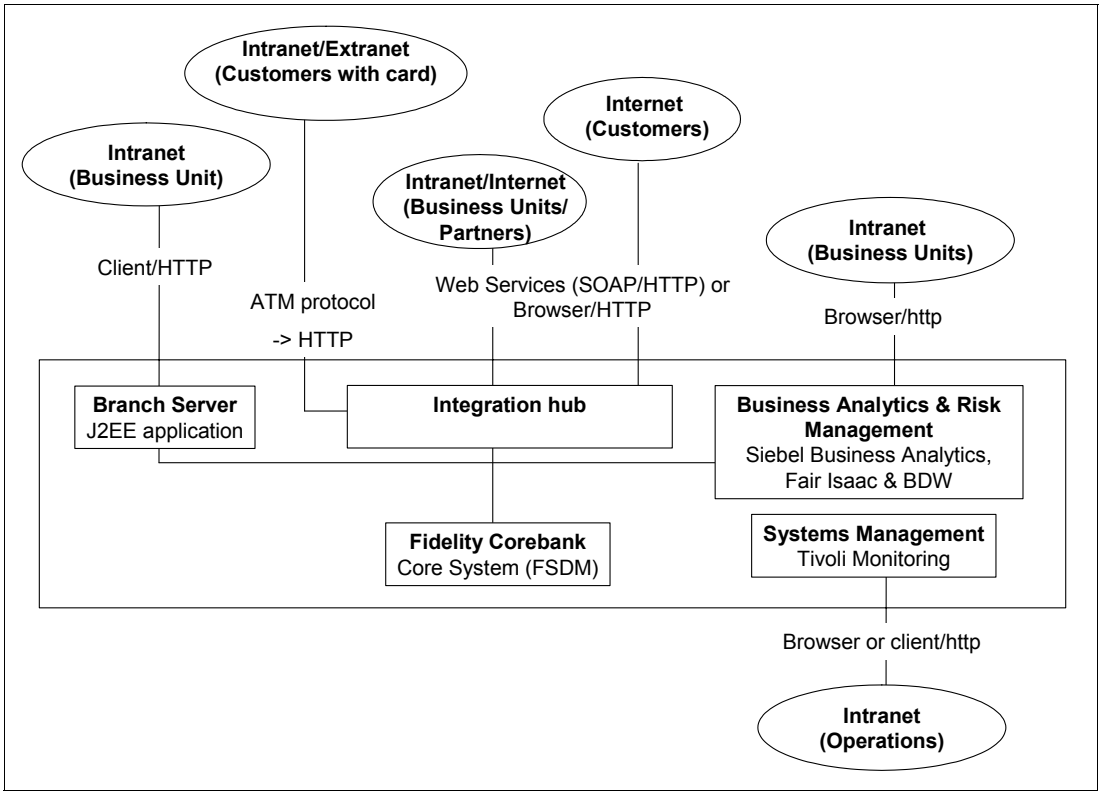


Figure 1-5 Smart Bank showcase architecture with emphasis on the channels that are discussed

Business intelligence

Based on the extract, transform, and load (ETL) process from our core system operational database, a subset of the clients was loaded to an analytical database, which enriched the data within. This created our banking data warehouse on DB2 for z/OS, the same platform as our operational database. The banking data warehouse model is designed specifically for analytics and business intelligence. We then created an interface to Fair Isaac to further enrich our analytical database with client credit risk ratings and the probability of default ratings. This was run historically to create the necessary history for Basel II reporting. Finally, Siebel Analytics was used as the business intelligence engine to inquire on our banking data warehouse in order to try and identify key trends and client segments. From an architectural point of view, two key considerations helped us in our choice:

- We could use open standards to access the data (Open Database Connectivity)
- We could use the central analytical database that we chose based on its use of the key retail banking model (Information FrameWork)

Near real-time update of analytical database

Our GBS colleagues at IBM identified early on in the project that retail banks are striving for more up-to-date information feeds into their analytical databases for businesses to make more informed decisions based on near real-time information. Some analysis, such as fraud detection and money laundering, must be computed real time, with special mechanisms put

in place. Other forms of analysis such as the credit risk analysis that is discussed in the proof point “Risk and compliance” on page 21, do not require up-to-date information because they are primarily historical in nature. However, most other forms of business analytics benefit from increased real-time data feeds.

There are many possible methods of achieving this. We chose two from a demonstration point of view. They are:

- CICS Business Event Publisher
- WBISF and the SOA model

Both use the asynchronous update mechanism afforded by WebSphere MQ. To ensure a minimal impact on the quality of service (response time) that our clients (channels) experience, we chose an asynchronous update mechanism to collect the information and quickly register the update to the banking data warehouse by writing this information to a queue. This will be present in the same unit of work as the channel transaction. A second unit of work under a different service class (quality of service ranking that is lower and not as critical) then picks up the message and updates the banking data warehouse when there are sufficient machine resources to do so. During the demonstration at high volumes, this remains virtually real time. Figure 1-6 shows the system context diagram and illustrates how these concepts are used in the integration hub in order to integrate with the analytical database as well. This system context diagram looks a little more like the IBM SOA reference diagram that we used as a base for our SOA model diagram in the Tivoli Enterprise Portal.

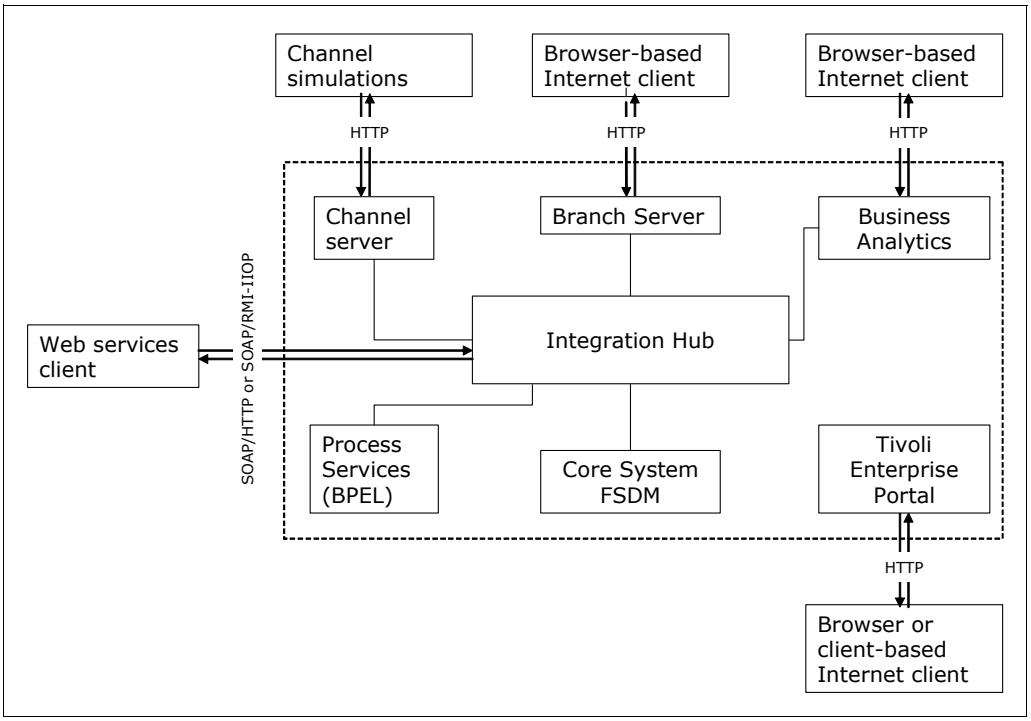


Figure 1-6 System context diagram

Risk and compliance

Fair Isaac's Triad product provided us with the necessary risk scoring. One of the key principles that was used here was to reuse the existing business intelligence infrastructure. We reused the banking data warehouse model to store the data. We used our existing Siebel Business Analytics dashboards that also provided a more visual demonstration mechanism as the reporting mechanism. We then created an entire range of Basel II and risk analysis reports based on the historical data in our database. This allowed us to demonstrate an aspect of credit risk compliance.

Business continuity and high availability

Another aspect of risk and compliance is minimizing operational risk. There are many reports under Basel II that cover this area, but we focused on the infrastructure and built our operational environment to be resilient and highly available using many of the technologies available to us through IBM System z0-109, including the following:

- Geographically Dispersed Parallel Sysplex™ (GDPS®) HyperSwap™ for disk resiliency
- Parallel Sysplex for server resiliency
- Sysplex Distributor for workload distribution and high availability
- Virtualization to allow dynamic resource allocation and sharing of resources
- HiperSockets™ and virtual local area network (VLAN) for virtualization of the network within the IBM System z9-109 machine
- Workload Manager (WLM) for intricate workload management on z/OS for service-level agreements and machine priorities for z/VM

Branch transformation and IT simplification

Our branch network forms a part of the multichannel architecture and adheres to the same principles. The branch server is a J2EE application, and consequently, adheres to the principles of being able to run on any platform that can run the Java Virtual Machine (JVM™). We leveraged this capability and deployed the branch servers to a centralized platform on IBM System z9-109, benefiting from the resiliency and availability discussed in "Business continuity and high availability" on page 21, and the cost efficiency of a highly simplified infrastructure and network. We chose Linux running as a guest under z/VM as the deployment platform for the combined horizontal and vertical scalability.

This also provided us the flexibility to deploy the same branch server to other platforms, depending on the business requirements and qualities of service required. We chose the BladeCenter environment to represent a regional hub or a slightly different business channel that uses the same branch server functionality. This server also ran on Linux, but on Intel® blades.

The main reason why clients are pursuing this sort of infrastructure simplification is because of the cost-efficiency it enables. The second reason is the flexibility this model affords. Figure 1-7 illustrates the physical deployment environment for the Smart Bank showcase.

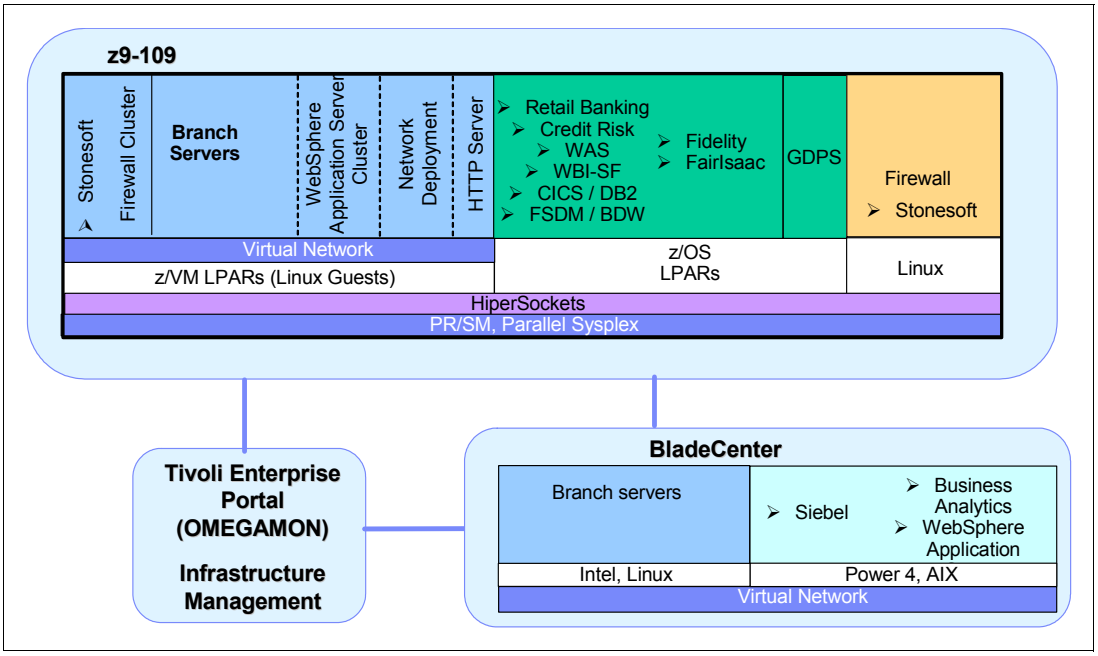


Figure 1-7 Architecture overview: Physical view

In the architecture diagrams, we called the integration techniques, protocols, and mechanisms used in the project as the *Integration Hub*. This uses many of the components such as WebSphere MQ that form the basis of the *Enterprise Service Bus* (ESB) and are available as a physical implementation with WebSphere V6, which we had not installed at this stage of the Smart Bank showcase.

1.3.2 Systems management requirements

The showcase requirements for systems management are listed here:

- The Smart Bank project requires an Enterprise view of its systems across different platforms for it to monitor and manage the system using one point of control
- Aligning the business goals setup with the IT infrastructure using WLM and IBM Tivoli Monitoring V6 to monitor the quality of service that is delivered to our clients, business units, and partners.
- As per the Smart Bank project goals, demonstrate the business proof points through the TEP using the Logical and Physical views in order to show the bank running at full volume and handling the use cases thrown at it during different times in a day.
- Measuring the following:
  - Operational online and batch throughput
  - Response time across our channels to clients and business units
  - Utilization of our IT resources, primarily CPU, but also I/O, and memory
  - Monitoring and managing systems availability and pinpointing problem areas
  - Demonstrating the scalability through TEP and showing how IBM Tivoli Monitoring can assist in this aspect

- Showing control of the environment from a single point. Although automatic processes adjust the system response based on the business parameters, there are occasions when direct manual action is required and the environment adjusted, as with the On/Off Capacity on Demand scenario. This is required because this demonstrates how flexible the infrastructure is, and despite being virtual, is capable of quickly responding to the demands of the business.

1.3.3 Use cases this IBM Redbook focuses on

The Smart Bank showcase lays out a number of business proof points. To demonstrate these, we created a number of use cases that simulate events at different times of the day in the bank. We deliberately decided to focus on a few of the more important use cases in order to be able to discuss the Logical views we created in TEP. The use cases have also been chosen to cover a cross-range of the IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6 capabilities.

Following are the use cases this book focuses on from the Smart Bank showcase proof points perspective:

- Core system transformation (SOA), described in 5.1, “Core system transformation (service-oriented architecture)” on page 258.
- Optimization of IT resources (On/Off Capacity on Demand), described in 5.2, “Optimization of IT resources (On/Off Capacity on Demand)” on page 272.
- Branch transformation and infrastructure simplification (heterogeneous view), described in 5.3, “Branch transformation and infrastructure simplification (heterogeneous view)” on page 280.

We discuss these use cases after discussing monitoring architecture, implementation, and Physical and Logical views.

1.4 IBM Tivoli Monitoring: Architectural design and description

The Smart Bank showcase is focused on IBM System z, as are the bank’s employees. The employees were therefore, looking for a reliable system management tool that supports their requirements in two different ways:

- Showing application-related Logical views, where the health of the application-supporting components can be shown
- Identifying and analyzing the root cause of a slowdown or outage in one of the applications

What is required is a solution that supports both these requirements and links them both neatly.

To fulfill the Smart Bank showcase requirements, we select several products from the Tivoli portfolio. After selecting the products, the different alternatives are discussed, pin-pointing to the following:

- The products selected
  - The components involved
  - The relationship between various products
  - The communication paths

- The architectural reflection
  - High availability
  - Embedding in existing environments
  - Virtualization

In order to make the different alternatives visible, the Tivoli product components that are to be picked, how they fit together, and the platforms on which they are supported, are discussed first.

This is followed by a discussion of different solutions, introducing the idea of virtualizing the Tivoli Enterprise Portal Server. The different approaches are then compared.

1.4.1 Products selected from the Tivoli portfolio

The project uses two main products from the Tivoli portfolio:

- IBM Tivoli Monitoring V6

  This is the new IBM Tivoli Monitoring solution released in October 2005. It offers a monitoring solution for large numbers of platforms and applications, when integrating all the information that is gathered, into a common infrastructure.

  The IBM Tivoli Monitoring V6 portal enables the consolidation and presentation from different information sources, including the IBM Tivoli Monitoring agents on distributed IBM Tivoli OMEGAMON agents on z/OS, and application data from IBM Tivoli Composite Application Manager V6. IBM Tivoli Monitoring V6 may then be used to create self-defined Logical views, where the health of an application and its physical implementation can be visualized.

- IBM Tivoli Composite Application Manager V6

  This was introduced in November 2005. This is a set of products that addresses the necessity for service and application monitoring, while fully integrating into the IBM Tivoli Monitoring V6 infrastructure. Beside this, it adds powerful functions to other interested groups such as application developers, with supporting deep-dive analysis for transactions and response times.

Together, these two products gather all the data that is required for performance measurement and bottleneck identification in the Smart Bank showcase environment.

IBM Tivoli Monitoring V6

IBM Tivoli Monitoring V6 is an enterprise-class, easy-to-use solution that optimizes the performance and availability of your entire IT infrastructure. Through a single customizable workspace portal, you can proactively manage the health and availability of your IT infrastructure end-to-end, including operating systems, databases, and servers across distributed and host environments. IBM Tivoli Monitoring detects bottlenecks and potential problems in essential system resources, and helps you to automatically recover from critical situations, ensuring that your business-critical applications are up and running.

Built on a lightweight, highly scalable architecture, IBM Tivoli Monitoring is quickly deployed and easily managed for quick time-to-value and low cost of ownership. Simple and centralized control, enhanced visualization of information, ease of use, and historical and real time reporting allow you to quickly access the information you require in customizable formats in order to rapidly identify, diagnose, and resolve situations.

By linking IT services to processes, data, skills, and tools through its single user workspace, IBM Tivoli Monitoring allows you to view consistent data across technology domains, and align IT services with business goals to deliver immediate value. IBM Tivoli Monitoring, when

combined with IBM Tivoli's composite application, event, network, and service-level management solutions, offers complete and integrated availability management, ensuring that your IT resources and staff are operating efficiently and effectively in alignment with your business requirements and priorities.

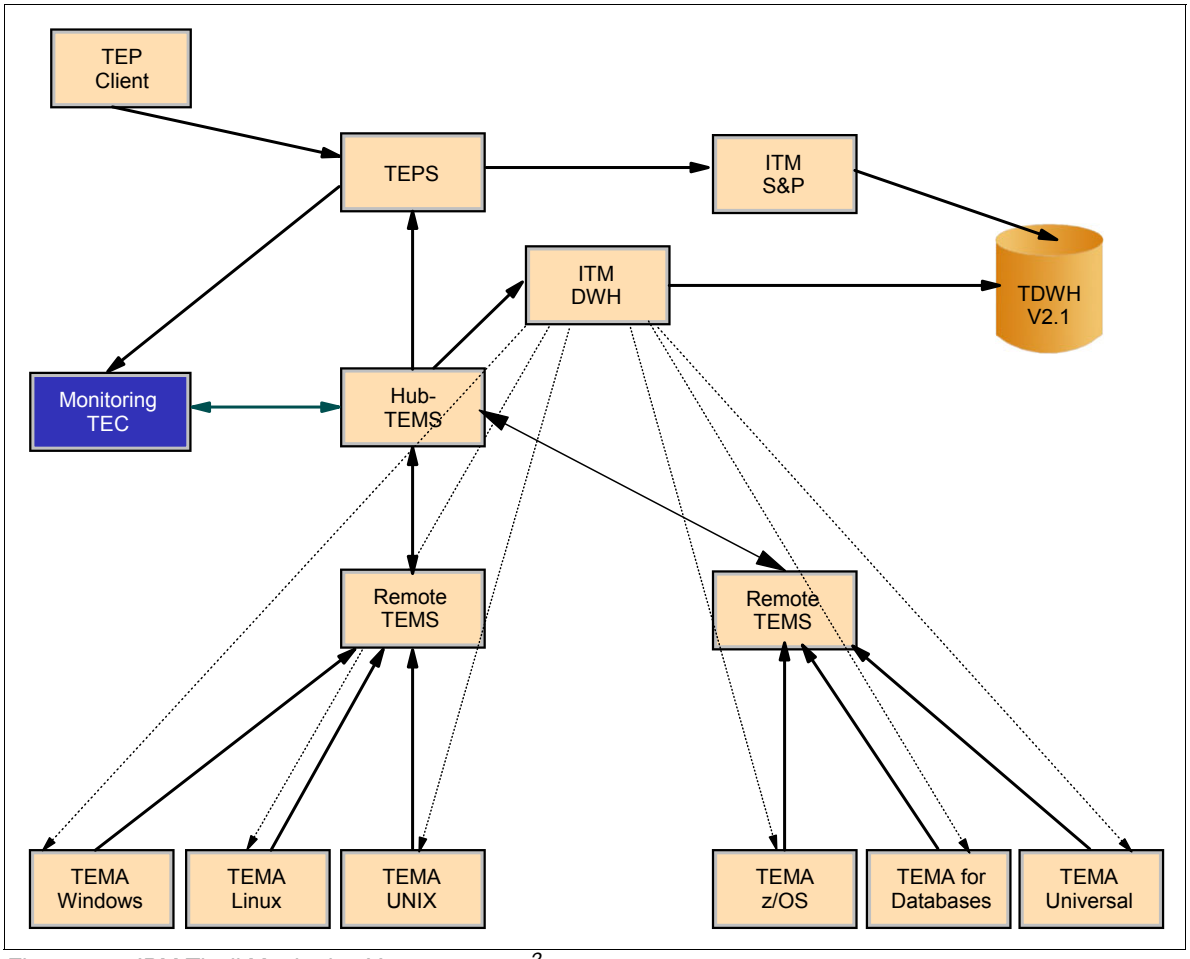


Figure 1-8 IBM Tivoli Monitoring V6 components[2]

IBM Tivoli Monitoring V6 comprises the following multiple components, as shown in Figure 1-8.

- IBM Tivoli Enterprise Managing Server
- IBM Tivoli Enterprise Portal Client
- IBM Tivoli Enterprise Portal Server
- IBM Tivoli Enterprise Managing Agent
- IBM Tivoli Data Warehouse V2.1
- IBM Tivoli Enterprise Console

[2] Although not shown in Figure 1-8, the Tivoli Enterprise Managing Agents have the ability to connect to the Tivoli Enterprise Managing Server Hub and are not required to go through the Tivoli Enterprise Managing Server Remote.

The following sections provide a brief description about each of these components. For detailed descriptions about the IBM Tivoli Monitoring V6 components, refer to *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143. To get up-to-date information about the platform support matrix for IBM Tivoli Monitoring V6.1, visit the following Web site:

http://www-306.ibm.com/software/sysmgmt/products/support/Tivoli_Supported_Platforms.html

Tivoli Enterprise Managing Server

The Tivoli Enterprise Managing Server is the most important component within the IBM Tivoli Monitoring Services foundation. All the other architectural components depend directly on the way this functions. The Tivoli Enterprise Managing Server acts as a collection and control point for alerts received from agents, and collects their performance and availability data.

The Tivoli Enterprise Managing Server is responsible for tracking the operability of the connected components by exchanging heartbeat requests with all the Tivoli Enterprise Management Agents connected to it.

There is a difference between *Tivoli Enterprise Managing Server Hub* and *Tivoli Enterprise Managing Server Remote*. Any Tivoli Monitoring infrastructure requires at least one Hub Tivoli Enterprise Managing Server. The Tivoli Enterprise Managing Server Remote is used to meet scalability requirements. The Tivoli Enterprise Managing Server Hub stores, initiates, and tracks all the situations and policies, and is the central repository for storing all active conditions. The Tivoli Enterprise Managing Server storage repository is a proprietary database format grouped as a collection of files located on the Tivoli Enterprise Monitoring Server.

Following are the platforms that are supported:

- z/OS
- AIX
- Linux
- Solaris™
- Windows®

Tivoli Enterprise Portal Client

The Tivoli Enterprise Portal Client is a Java-based user interface that connects to the Tivoli Enterprise Portal Server to view all the monitoring data collections.

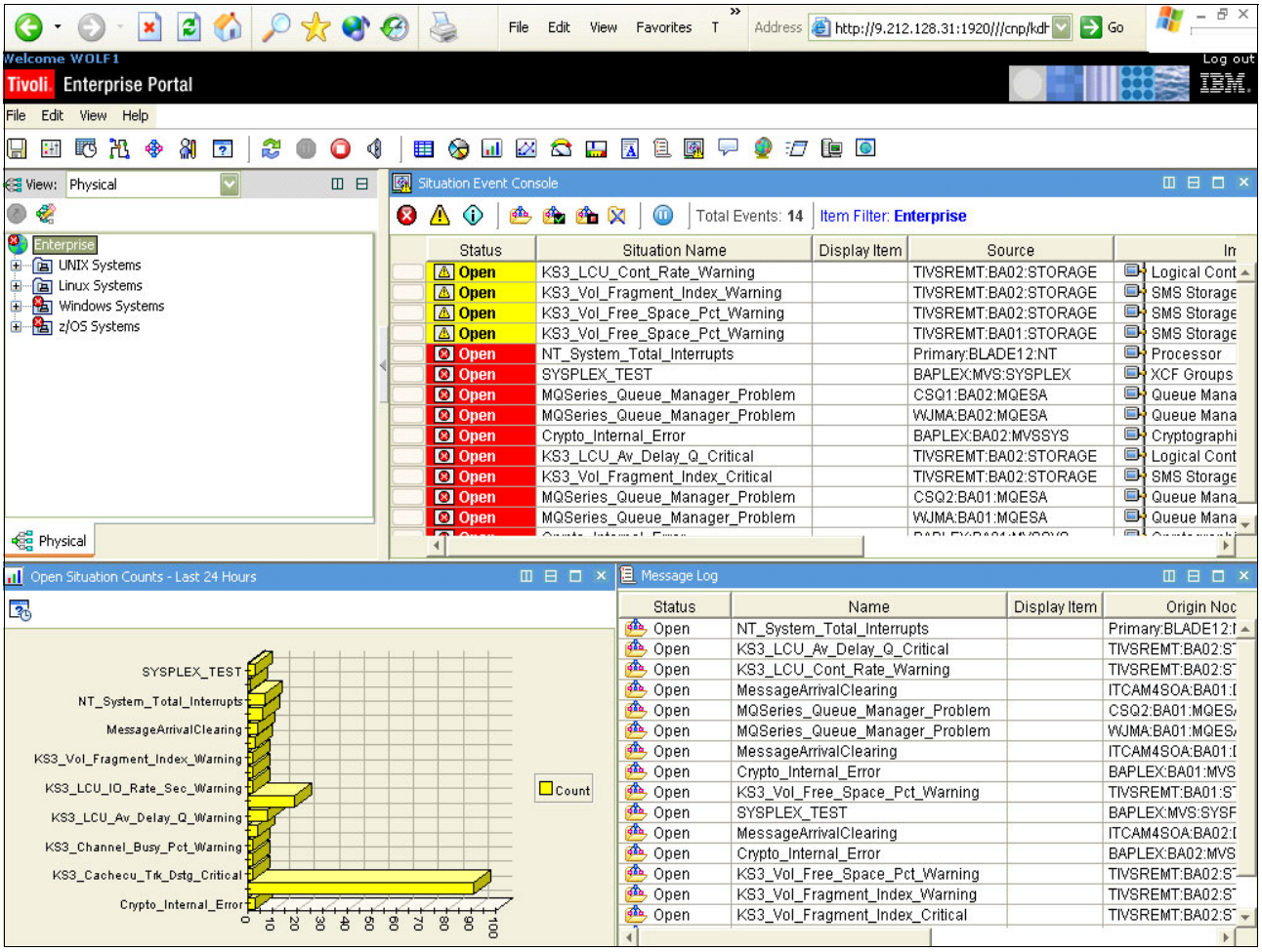


Figure 1-9 The default TEP Client Welcome workspace, when the distributed environment component is installed

Figure 1-9 shows the predefined workspace displayed by default after a successful login to the Tivoli Enterprise Portal Server. The Tivoli Enterprise Portal Client may run as a Java application within the Microsoft® Internet Explorer®, or as a fat Java application on Linux or on Microsoft Windows.

The Tivoli Enterprise Portal Client may display information from all the monitored systems and applications that are supported by IBM Tivoli Monitoring and IBM Tivoli OMEGAMON. For a detailed list, visit the following Web site:

`http://www-306.ibm.com/software/tivoli/sw-atoz/indexM.html`

Tivoli Enterprise Portal Server

The Tivoli Enterprise Portal Server is the central component for mapping the technical information gathered by the agents and the Tivoli Enterprise Managing Server to match client requirements and enable the presentation of this infrastructure to the user.

The Tivoli Enterprise Portal Server is the repository for the entire graphical presentation of monitoring data. All the working data is stored in a relational database, including user-specific user rights and views. The Tivoli Enterprise Portal Server provides the core presentation

layer, allowing the retrieval, manipulation, analysis, and preformatting of data. The Tivoli Enterprise Portal Server maintains a persistent connection to the Hub Tivoli Enterprise Managing Server, and can be considered a logical gateway between the Hub Tivoli Enterprise Managing Server and the Tivoli Enterprise Portal Client.

A relational database management system must be installed on the same physical system prior to the Tivoli Enterprise Portal Server installation. This database stores all the presentation data and manipulation rules. Also, user settings, if any, are stored in this database.

Following are the platforms that are supported:

- Linux on IBM System x
- Linux on IBM System z
- Windows
- (AIX will be supported in the next release)

Tivoli Enterprise Managing Agent

The Tivoli Enterprise Managing Agent is installed and implemented on the system or subsystem that requires data collection and monitoring. It is responsible for gathering the required data. It does this by sampling specific attribute groups that are inherent to a given agent, on request by an user or triggered by a given situation.

In the Smart Bank project, the following agents are used:

- IBM Tivoli OMEGAMON XE on z/OS V3.10
- IBM Tivoli OMEGAMON XE for DB2 on z/OS V3.10
- IBM Tivoli OMEGAMON XE for Mainframe Networks V3.10
- IBM Tivoli OMEGAMON XE for CICS on z/OS V3.10
- IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1.0
- IBM Tivoli Monitoring for distributed systems on Windows V6.10
- IBM Tivoli Monitoring for distributed systems on AIX V6.10
- IBM Tivoli Monitoring for databases
- IBM Tivoli Monitoring for Composite Application Manager Integration
- IBM Tivoli Monitoring for Service-Oriented Architecture (SOA)

Tivoli Data Warehouse V2.1

Tivoli Monitoring V6 is delivered with a new version of the Tivoli Data Warehouse. The new Tivoli Data Warehouse V2.1 is based on a relational database system of your choice, including IBM DB2 Universal Database™.

The Tivoli Monitoring agent for data warehousing is responsible for transporting all the gathered data to the data warehouse. Whenever an application exports data to the database for the first time, it automatically generates new tables. There are no pre-existing tables for any application, with the exception of two tables that Tivoli Data Warehouse generates to collect information regarding the data being exported from other applications. The data collection is initiated by the Tivoli Data Warehouse agent directly with the product-specific agent (Tivoli Enterprise Management Agent for Windows, UNIX, and so on).

The behavior of historical data collection is controlled by the Tivoli Enterprise Portal Server that stores the following parameters for the Tivoli Monitoring Agent for Data Warehousing in the Tivoli Enterprise Managing Server:

- The attribute groups to be collected
- The frequency at which to collect data
- The place where this data is to be stored temporarily
- The periodicity at which to transfer the temporary data to the Tivoli Data Warehouse

The Summarization and Pruning Agent is responsible for maintaining the warehouse functions inside the database in order to perform the required aggregation, summarization, and pruning steps. Through Tivoli Enterprise Portal Server, the following settings are applied:

- The period for which raw data is to be retained
- Decision about creating hourly, daily, monthly, quarterly, and/or yearly reports
- The timeframe for which these generated reports are to be maintained

The Summarization and Pruning Agent ensures that the Tivoli Data Warehouse database does not grow in an unlimited manner.

Restriction: The Summarization and Pruning Agent currently does not support the IBM Tivoli OMEGAMON products. This means that all z/OS-based Tivoli OMEGAMON products may report data to the data warehouse, but the maintenance of the data is currently not done by the Summarization and Pruning Agent. Manual intervention is required. Otherwise, your database will grow in an unlimited manner.

This restriction will be removed when V4.1 becomes available.

Tivoli Enterprise Console event synchronization

The Tivoli Enterprise Console event synchronization enables a two-way synchronization between the Tivoli Monitoring V6 infrastructure and the Tivoli Enterprise Console. The Tivoli Enterprise Console event synchronization component sends back to the monitoring server, the updates of situation events that are forwarded to the Tivoli Enterprise Console server. Actions performed in the Tivoli Enterprise Console for Tivoli Monitoring V6 situations are reflected in the Tivoli Enterprise Portal Server, and then in the Tivoli Enterprise Portal Client.

IBM Tivoli Composite Application Manager

As described in 1.1.4, “The banking sector’s future key business drivers” on page 7, clients have a specific requirement to monitor their WebSphere environment, the application server, IBM WebSphere MQ, and the objects inside. Additionally, SOA-driven applications are also monitored.

The scope of these functions and the related products are included in the IBM Tivoli Composite Application Manager.

The following products are used in the Smart Bank project:

- IBM Tivoli Composite Application Manager for WebSphere Application Server
- IBM Tivoli Composite Application Manager for SOA
- IBM Tivoli OMEGAMON XE for WebSphere Application Server on z/OS

IBM Tivoli Composite Application Manager for WebSphere

IBM Tivoli Composite Application Manager for WebSphere results from the merger of the following two standalone products into one:

- IBM WebSphere Studio Application Monitor
- IBM Tivoli OMEGAMON XE for WebSphere Application Server on z/OS

These products now share a common data collector to get the required data from the monitored WebSphere Application Server.

Figure 1-10 illustrates the components, and how they integrate with IBM Tivoli Monitoring V6.

IBM Tivoli Composite Application Manager for WebSphere comprises three major components:

- IBM Tivoli Composite Application Manager/WebSphere Application Server managing server

  The managing server is the central point of IBM Tivoli Composite Application Manager/WebSphere Application Server infrastructure, where deep-dive analysis for the monitored application can be carried out. This component is split up into several subcomponents, including a WebSphere Application Server, in order to host an application that is accessible to the users.

- Tivoli Enterprise Management Agent for IBM Tivoli Composite Application Manager/WebSphere Application Server

  The Tivoli Enterprise Management Agent for IBM Tivoli Composite Application Manager/WebSphere Application Server is an agent that is fully integrated into the IBM Tivoli Monitoring V6 system management infrastructure. It gets all the required data from the data collector running inside the WebSphere Application Server instance.

- IBM Tivoli Composite Application Manager/WebSphere Application Server data collector

  The data collector for IBM Tivoli Composite Application Manager/WebSphere Application Server runs inside each monitored WebSphere Application Server instance. It reports its data to the managing server and the Tivoli Enterprise Management Agent, if present.

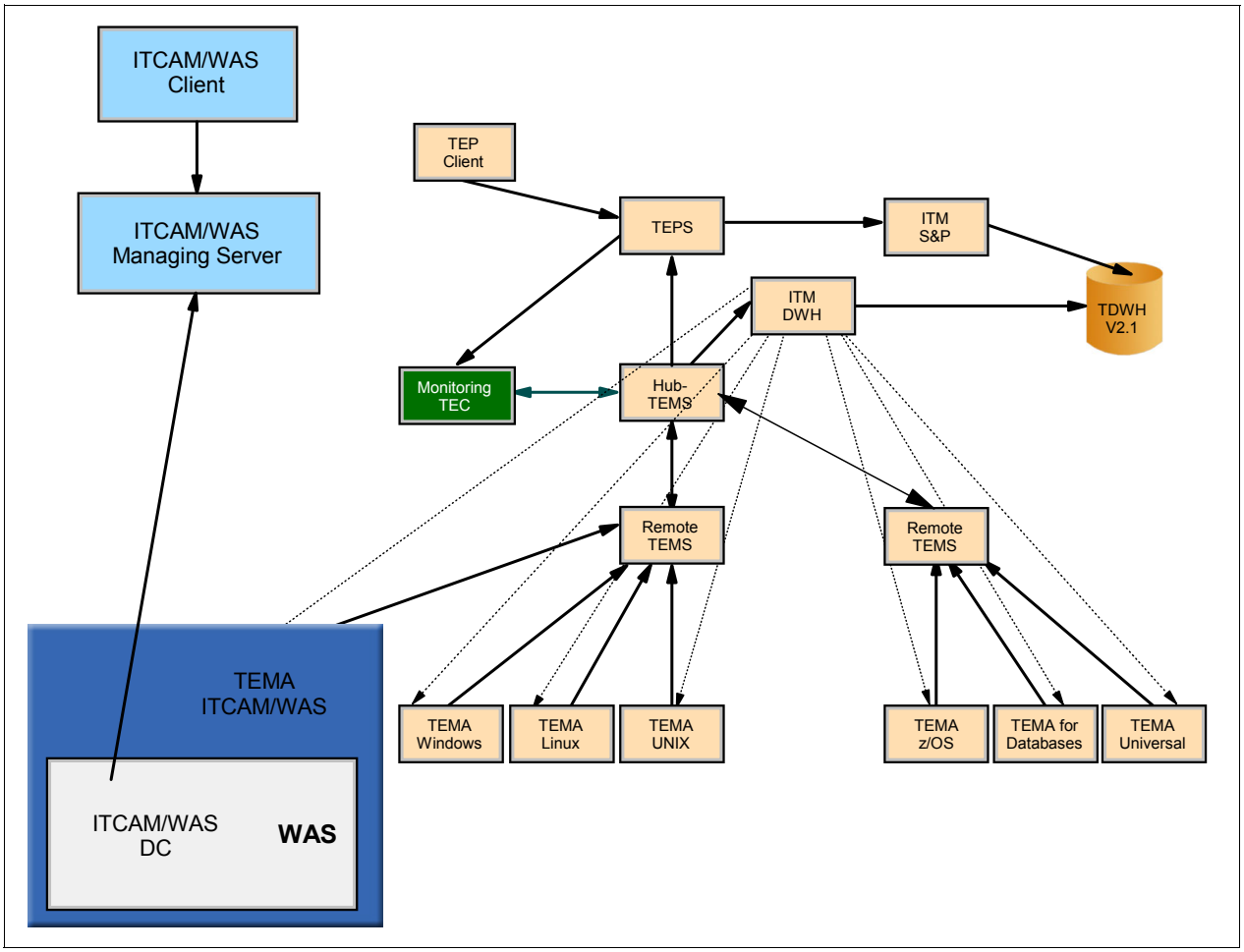


Figure 1-10 IBM Tivoli Composite Application Manager for WebSphere and its components

In this book, only the use of the Tivoli Enterprise Management Agent for IBM Tivoli Composite Application Manager/WebSphere Application Server is described. Deep-dive analysis of WebSphere Application Server is not covered. For a detailed description of IBM Tivoli Composite Application Manager/WebSphere Application Server, refer to *IBM Tivoli Composite Application Manager V6.0 Family: Installation, Configuration, and Basic Usage,*, SG24-7151.

IBM Tivoli Composite Application Manager for Service-Oriented Architecture

IBM Tivoli Composite Application Manager for SOA manages Web services. These services are remote processing facilities that are defined through the use of Web Services Definition Language (WSDL). These services are accessed by using SOAP over HTTP. Internally, Web services are implemented using Java API for Extensible Markup Language-based Remote Procedure Call (JAX-RPC). IBM Tivoli Composite Application Manager for SOA installs itself as the JAX-RPC handler in order to capture and manage Web service calls.

IBM Tivoli Composite Application Manager for SOA consists of the following logical components:

- A Web services data collector within the Tivoli Enterprise Management Agent

  The monitoring agent is installed locally on every application server environment where Web services are to be managed. It intercepts and instruments Web service requests and responses. The data collected by the monitoring agent about Web services is written and stored in a log file. Additionally, it is sent on request to the IBM Tivoli Monitoring V6 infrastructure.

- An Eclipse-based viewer that processes the log files that are generated by the Web services data collector

  This generates visual representations of the various characteristics of the monitored Web services.

Figure 1-11 references the components that come with IBM Tivoli Composite Application Manager for SOA.

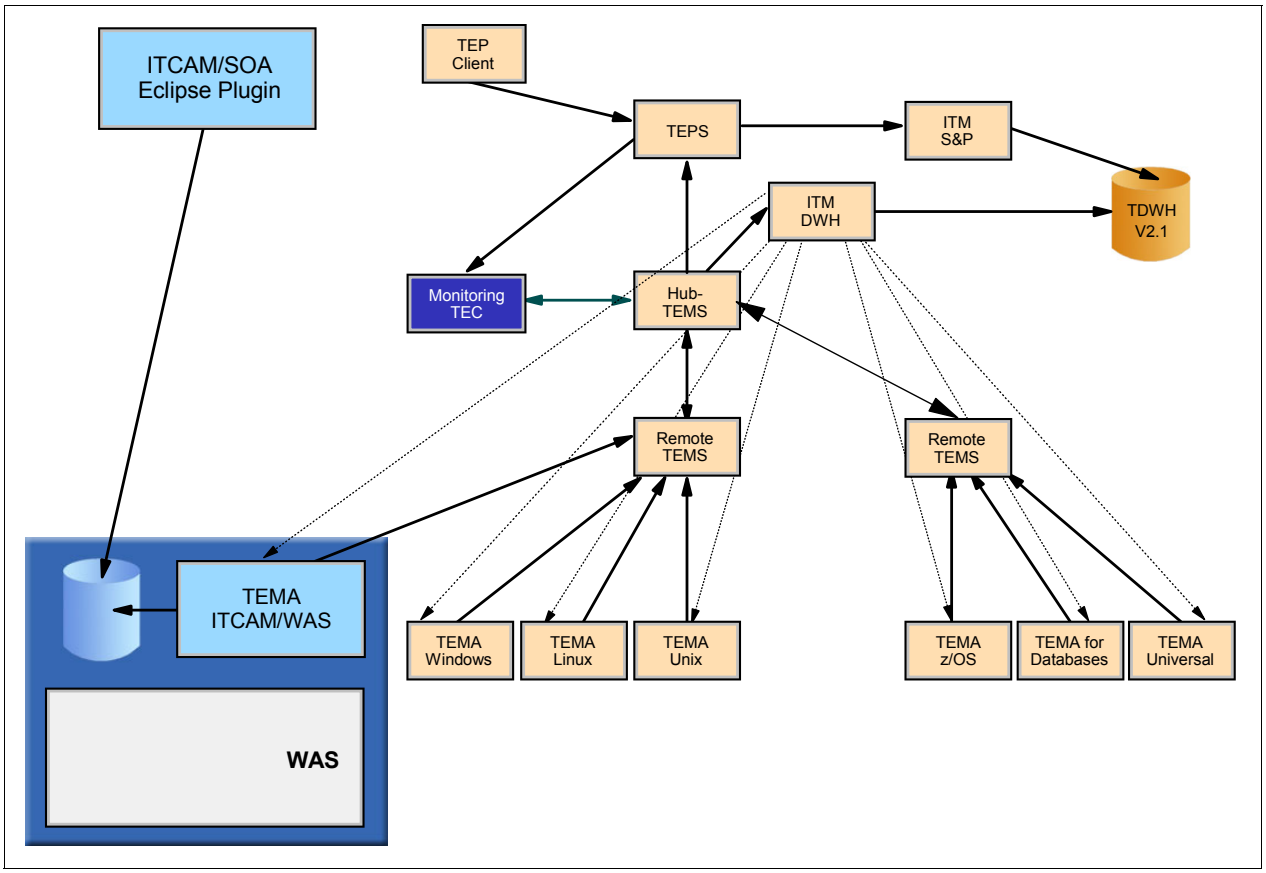


Figure 1-11 IBM Tivoli Composite Application Manager for SOA and its components

The Tivoli Enterprise Management Agent is added on each of the systems where SOA requests are to be measured. The agent connects to the IBM Tivoli Monitoring V6 infrastructure. Through this connection, it may leverage services such as setting thresholds, storing historical data, data warehousing, and so on.

The focus of this project is to generate Logical views pertaining to the Smart Bank showcase and system infrastructure. Therefore, the Eclipse-based viewer is not described in detail.

1.4.2 General architectural considerations

After identifying all the components that are necessary to get the required data, it is important to set up a reliable IT system management infrastructure, where the collected data is consolidated and grouped into Logical views.

The Smart Bank team requested an infrastructure that withstands any major outage. This means that we keep in mind the fact that any component may fail at any time, and be prepared for this. If a server or its connection to the network is lost, an alternative must be available.

In our Smart Bank project, we assume that if a software component fails, it can be restarted. If this does not happen, we conclude that the server is lost completely.

The following guidelines have therefore been put in place for the subsequent alternatives:

- The Tivoli Enterprise Management Server Hub must be able to recover in 10 minutes or less
- On each of the systems where a Tivoli Enterprise Management Server Remote resides, the installed agents report directly to that server. They do not have a second path to another Tivoli Enterprise Management Server. As mentioned earlier, if we cannot restart, the entire system is lost.
- Each agent on a Windows, or Linux on POWER® platform connects to the two Remote Tivoli Enterprise Management Servers located on Windows and on AIX:
  - The Windows agents use the Windows-based Tivoli Enterprise Management Server Remote as their primary Tivoli Enterprise Management Server
  - All the others agents use the AIX-based Tivoli Enterprise Management Server Remote as their primary Tivoli Enterprise Management Server

  This algorithm ensures workload balancing across our two Remote Tivoli Enterprise Management Servers.

  Note: In high-scaling environments, where a large number of agents (more than 200 agents) connect to one Remote Tivoli Enterprise Management Server, it is useful to set up a dedicated backup Tivoli Enterprise Management Server Remote that is used only when the primary Tivoli Enterprise Management Server Remote fails.

- z/OS assumptions
  - On each z/OS partition, there is one Tivoli Enterprise Management Server Remote at all times
  - Between the Tivoli OMEGAMON components, z/OS internal communication is carried out only through the Systems Network Architecture (SNA).
  - All Tivoli OMEGAMON products running on z/OS communicate with their local Tivoli Enterprise Management Server Remote only. If they require communication with the Hub Tivoli Enterprise Management Server, it is carried out through the Remote Tivoli Enterprise Management Server.
- No agents are connected directly to the Tivoli Enterprise Management Server Hub at any time. Avoiding direct connections between the agents and the Tivoli Enterprise Management Server Hub keeps the data traffic to the Tivoli Enterprise Management Server Hub as low as possible. The restart time of the Tivoli Enterprise Management Server Hub after an emergency restart or switchover works faster because:
  - There are few data sets (files) to recover.
  - The agents' communication with the Tivoli Enterprise Management Server Remote is still up and running. Only a few Remote Tivoli Enterprise Management Servers must reconnect to the Hub Tivoli Enterprise Management Server.
  - If the Remote Tivoli Enterprise Management Server reconnects successfully, all the attached agents will go online at the same time.

  With this assumption, a well-organized communication structure within our system management infrastructure is ensured. If necessary, in order to meet targets, parameters in the Tivoli Enterprise Management Server Remote may be changed to speed up the process for reconnection, if the hub has failed.

Figure 1-12 shows the dependencies between the various IT components involved. *Appl.-Sys.* stands for the monitored application systems that will be loaded with the Tivoli Enterprise Management Server. At the very least, all of them will host an operating system agent. Additionally, they will have agents for the applications they host, such as DB2 Universal Database, Hypertext Markup Language (HTTP) Server, and so on.

Note the following points:

- All these agents connect to the appropriate primary and secondary Remote Tivoli Enterprise Management Server
- On z/OS, the agents connect only to the local Remote Tivoli Enterprise Management Server, using SNA
- The Tivoli Enterprise Management Server Remote is connected to the Hub Tivoli Enterprise Management Server

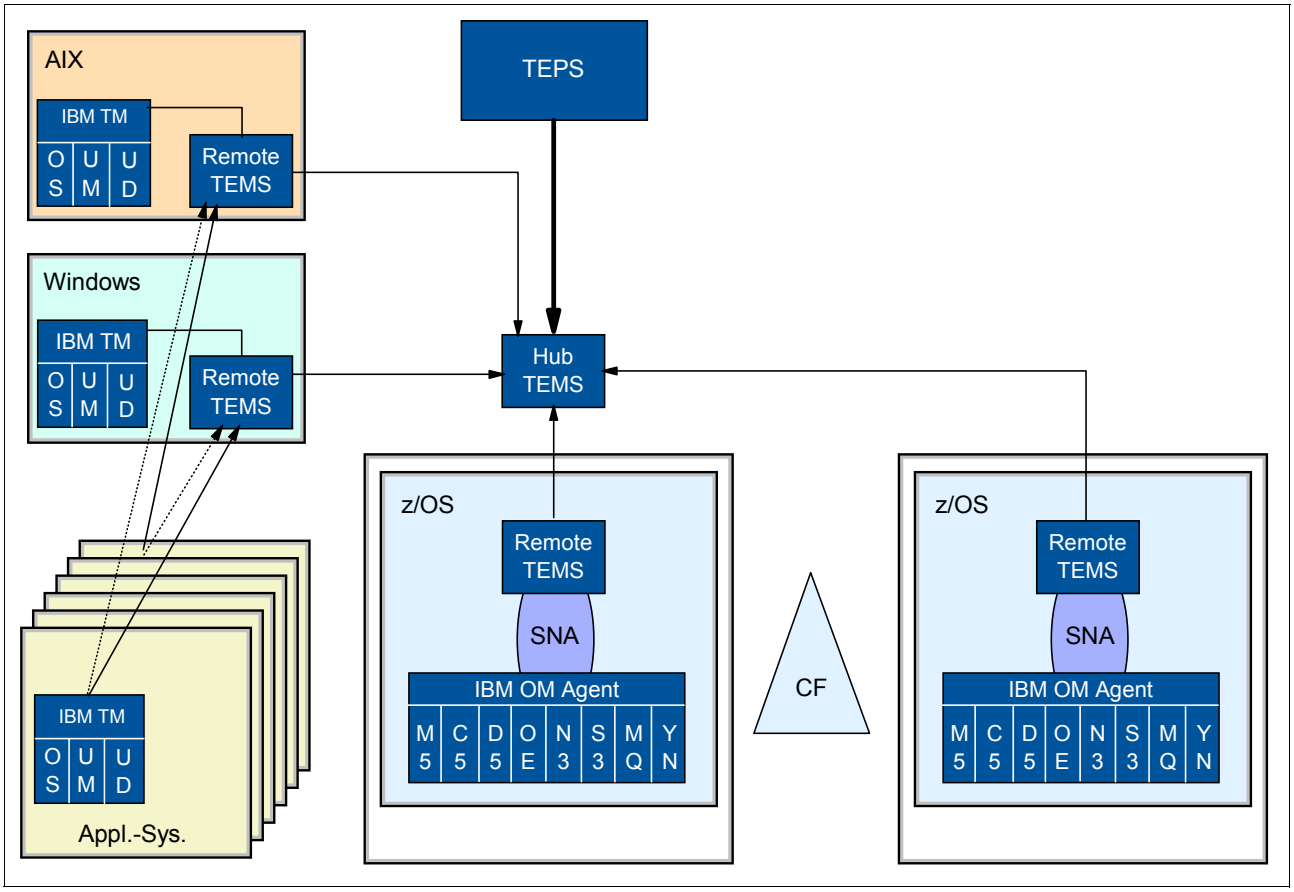


Figure 1-12 General IT infrastructure environment and the team's Tivoli environment assumptions

In the sections that follow, the different ways in which to place the two remaining components, the Tivoli Enterprise Management Server Hub and the Tivoli Enterprise Portal Server are described.

1.4.3 Distributed environment scenario

This scenario addresses a client situation involving a dominant distributed environment, and the z/OS system is one among the servers. The monitoring infrastructure is mostly driven by requirements, and is under the responsibility of those running the distributed environment. Due to this, the Tivoli Enterprise Management Server Hub and the Tivoli Enterprise Portal Server are placed on Windows. This implies that most of the maintenance must be performed on this platform. Figure 1-13 shows the suggested configuration.

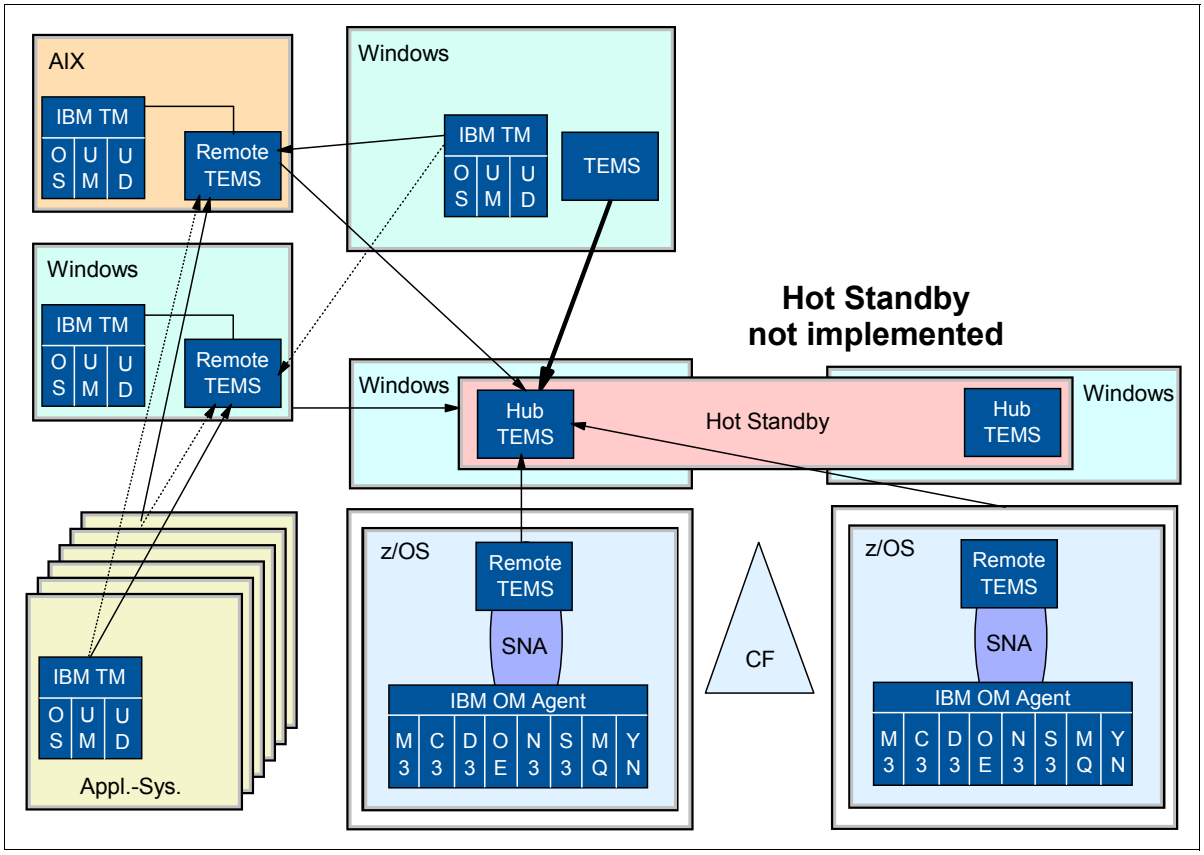


Figure 1-13 Windows-based IBM Tivoli Monitoring V6 system management infrastructure

Two different Windows systems are used to host Tivoli Enterprise Management Server and Tivoli Enterprise Portal Server. The infrastructure components for both Tivoli Enterprise Management Server and Tivoli Enterprise Portal Server must have powerful CPUs and plenty of memory. For details about the installation, refer to the following Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml

Note: You can also place the Tivoli Enterprise Management Server on AIX, Linux on IBM System x, or Sun™ Solaris. The Tivoli Enterprise Portal Server is also supported on Linux on IBM System x. Select your preferred platform. All the platforms support the essential features referenced in this book.

It is a good practice to place the Tivoli Enterprise Management Server and the Tivoli Enterprise Portal Server on different systems because both may require the same system resources at the same time. In the distributed area, you must pay special attention to network traffic. The Tivoli Enterprise Management Server must be able to handle all the communication across the entire infrastructure, including heartbeats, requests for data, and so on.

Following are the components:

- On Windows:
  - For Tivoli Enterprise Management Server Hub

    For the Tivoli Enterprise Management Server Hub on Windows (also UNIX and Linux) a feature called Hot Standby is available. As shown in Figure 1-13, the Tivoli Enterprise Management Server is the central point of communication. It is therefore important that this component is almost always up and running and delivers the service as expected. Even a single failure is not acceptable in a production environment.

    It is not possible to add a secondary Tivoli Enterprise Management Server Hub statement in the installation dialog. Manual changes to the z/OS configuration files are required. These changes have to be reinstated after each maintenance.
  - For Tivoli Enterprise Portal Server

    High availability functions and mirroring functions are not supported and the switch in the Hot Standby Tivoli Enterprise Management Server is not recognized.

    A detailed description about the workroom is documented in *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143.
  - Remote Tivoli Enterprise Management Server

    The Windows-based Tivoli Enterprise Management Server Remote acts as the Primary Tivoli Enterprise Management Server for all Windows-based agents. It also hosts all the distribution packages for the Windows systems.

    For Windows-based Remote Tivoli Enterprise Management Server, it is easy to add a secondary Tivoli Enterprise Management Server Hub statement. Thus, the support for the Hot Standby configuration of the Tivoli Enterprise Management Server Hub is fully supported
  - IBM Tivoli Monitoring agents

    On both the systems, IBM Tivoli Monitoring agents for monitoring the server software, such as the operating system and the DB2 Universal Database, will be installed. These agents connect to the Windows-based Tivoli Enterprise Management Server Remote as their primary server and to the AIX-based Tivoli Enterprise Management Server Remote as their secondary server. They thus have two ways of reporting to the Hub Tivoli Enterprise Management Server.
- On AIX
  - Remote Tivoli Enterprise Management Server

    The AIX-based Tivoli Enterprise Management Server Remote serves as the Primary Tivoli Enterprise Management Server for all AIX-based and Linux-based agents. It also hosts all the distribution packages for the AIX and Linux systems.

    For AIX-based Remote Tivoli Enterprise Management Server, it is easy to add a secondary Tivoli Enterprise Management Server Hub statement. Thus, the support for the Hot Standby configuration of the Tivoli Enterprise Management Server Hub is fully supported.

– IBM Tivoli Monitoring agents

The local agents (on this system, the operating system agent) connect only to the local Remote Tivoli Enterprise Management Server. If the machine is lost, which is one of the assumptions, a secondary connection will not help.

- On z/OS
  - Each logical partition (LPAR) hosts its own Remote Tivoli Enterprise Management Server.
  - All the z/OS-based agents connect only to their local Remote Tivoli Enterprise Management Server, using, in our case, SNA. No other protocol is configured.
  - Only the Tivoli Enterprise Management Server Remote can use IP communication to connect to the Hub Tivoli Enterprise Management Server.

1.4.4 IBM System z environment scenario

In this scenario, we address a client situation where we have a dominant IBM System z-based environment, and z/OS is the most important server. The monitoring infrastructure is mostly driven by the requirements, and is the responsibility of the people who handle the System z environment. Due to this, the Tivoli Enterprise Management Server Hub is placed on z/OS and the Tivoli Enterprise Portal Server is placed on Linux on IBM System z. This implies that most of the maintenance must be performed on z/OS and the Linux on IBM System z platform. Figure 1-14 shows the suggested configuration.

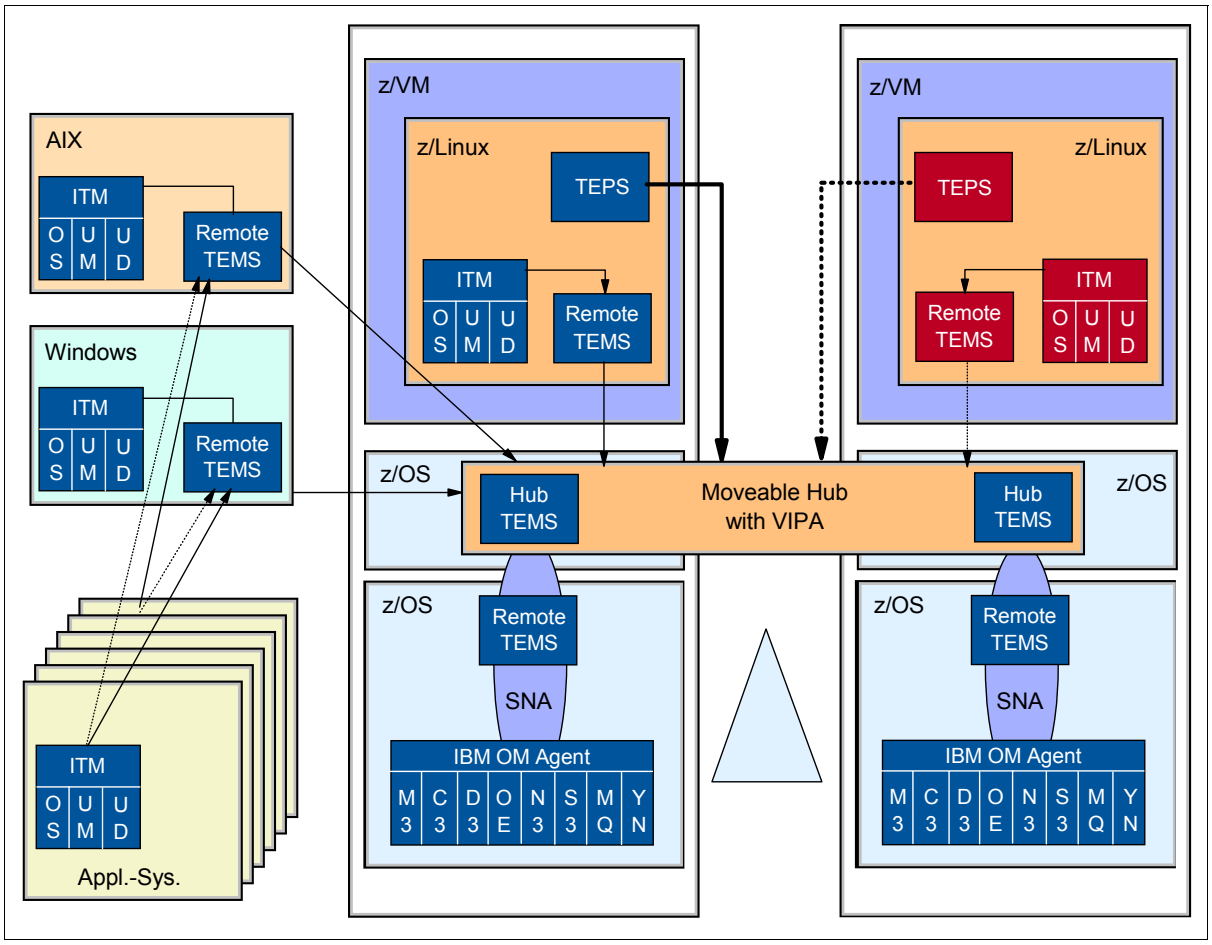


Figure 1-14 IBM System z-centric configuration

The components are:

- On Windows
  - Remote Tivoli Enterprise Management Server

    The Windows-based Tivoli Enterprise Management Server Remote acts as the Primary Tivoli Enterprise Management Server for all Windows-based agents. It also hosts all the distribution packages for the Windows systems.
  - IBM Tivoli Monitoring agents

    The local agents (on this system, the operating system agent) connect only to the local Remote Tivoli Enterprise Management Server. If we lose the machine, which is one of the assumptions, a secondary connection will not help.
- On AIX
  - Remote Tivoli Enterprise Management Server

    The AIX-based Tivoli Enterprise Management Server Remote serves as the Primary Tivoli Enterprise Management Server for all AIX-based and Linux-based agents. It also hosts all the distribution packages for the AIX and Linux systems.
  - IBM Tivoli Monitoring agents

    The local agents (on this system, the operating system agent) connect only to the local Remote Tivoli Enterprise Management Server. If the machine is lost, which is one of the assumptions, a secondary connection will not help.
- On z/OS
  - Moveable Hub Tivoli Enterprise Management Server

    On z/OS, the Tivoli Enterprise Management Server Hub feature, Hot Standby, does not exist at all.

    For the Tivoli Enterprise Management Server Hub on z/OS, a special solution, the Moveable Hub, is implemented by Tivoli services. The idea behind this is that on z/OS, the sysplex functionality enables functions such as Virtual IP Address (VIPA), which can also be used by Tivoli components such as Hub Tivoli Enterprise Management Server. For a detailed description, refer to “The Moveable Hub” on page 40.
  - Each LPAR hosts its own Remote Tivoli Enterprise Management Server. The communication with the Tivoli Enterprise Management Server Hub uses only SNA-based communication.
  - All the z/OS-based agents connect to their local Tivoli Enterprise Management Server Remote using only SNA. No other protocol is configured.
- Linux on IBM System z

  Linux on IBM System z runs under z/VM, the virtualization capability for the IBM System z platform. It enables the moving of a Linux on IBM System z image from one physical machine to another. All the addresses attached to that machine move with the virtual system. See “Tivoli Enterprise Portal Server implementation based on Linux on IBM System z under z/VM” on page 42 for more details.
  - IBM Tivoli Enterprise Portal Server

    Linux on IBM System z also gets its own Tivoli Enterprise Management Server Remote that serves as the primary (and only) Tivoli Enterprise Management Server for all Linux on IBM System z-based agents. It also hosts all the distribution packages for the Linux on IBM System z systems. The functions available are the same as that available

under Windows or Linux on Intel. The Tivoli Enterprise Portal Server does not support any high availability functions. The entire Linux on IBM System z image is now restarted on another physical system with the same addresses presented to the outside consumers.

This means that after restarting the Linux on IBM System z image, the connected user can continue to work without having to log in again when the system is moving. The Tivoli Enterprise Portal Server gives the impression of not being reachable. After initialization, the communication is reinstated automatically.

– Remote Tivoli Enterprise Management Server

The Linux on IBM System z also gets its own Tivoli Enterprise Management Server Remote that serves as the primary (and only) Tivoli Enterprise Management Server for all Linux on IBM System z-based agents. It also hosts all the distribution packages for the Linux on IBM System z systems.

Note: If there are enough Linux on IBM System z images available, place the Tivoli Enterprise Portal Server and the Tivoli Enterprise Management Server on different systems. Also, a secondary Linux on IBM System z-based Tivoli Enterprise Management Server Remote would be helpful to support the two different paths to the Hub Tivoli Enterprise Management Server.

In our case, we add Tivoli Enterprise Management Server to Linux on IBM System z in order to be able to use the HiperSockets for IP communication, which are available in System z. The Tivoli Enterprise Management Server running on this system can now connect to the infrastructure using only IP communication through the HiperSockets. This dramatically increases the performance of the components.

The Moveable Hub

The Moveable Hub implementation is documented in *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155. In the Smart Bank project, similar concepts are used. Only the paradigm that all z/OS internal communication is performed only through SNA, is added. Figure 1-15 describes the solution.

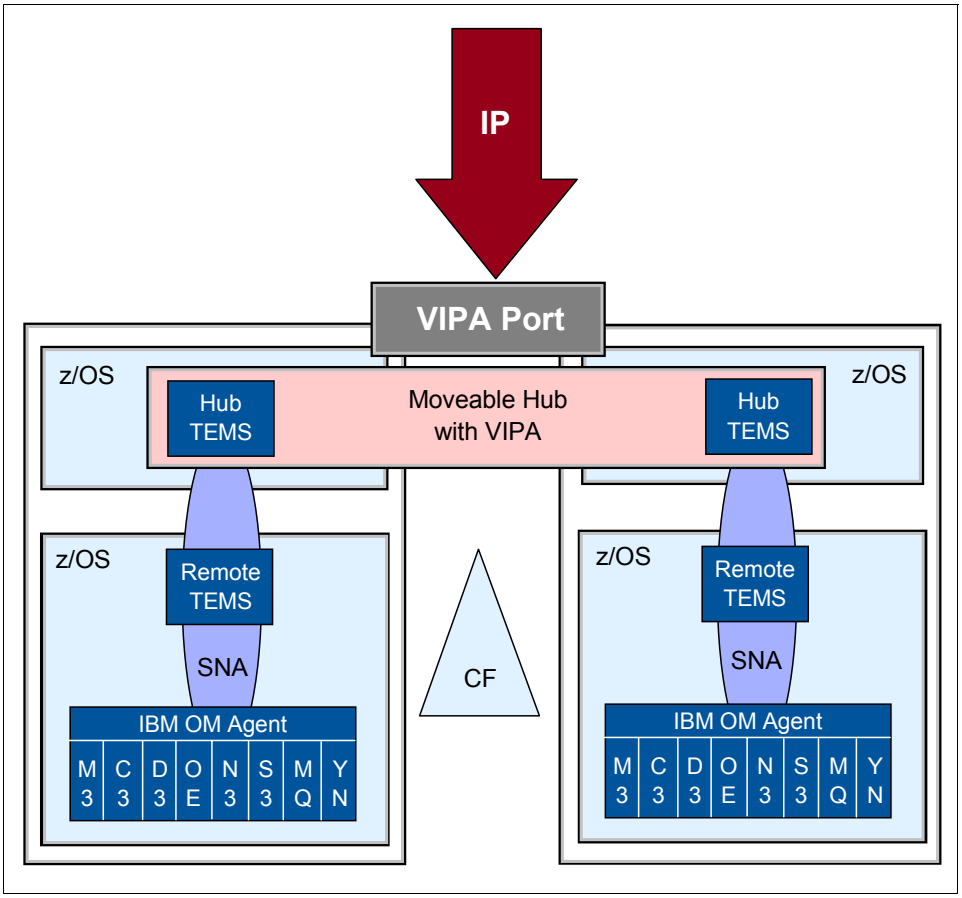


Figure 1-15 The Moveable Hub

As described earlier, the Tivoli Enterprise Management Server Hub is the core component of IBM Tivoli Monitoring V6, connecting the Tivoli Enterprise Portal Server with the underlying monitoring infrastructure such the Tivoli Enterprise Management Server Remote and the agents. It hosts all the logical monitoring data and its applied rules. If this component fails, the coordination inside the management infrastructure and the presentation to the user terminates.

The Moveable Hub implementation on z/OS allows the restart of the Tivoli Enterprise Management Server Hub on any member of the sysplex, assuming that common sysplex features such as disk sharing and so on are enabled.

Because the Tivoli Enterprise Management Server Hub no longer resides on a single physical or logical system, its communication credentials may change when it wants to connect to components outside the sysplex environment.

In the majority of cases, TCP/IP is the primary choice for communicating outside of z/OS. IP addresses are commonly bound to a specific machine so that a request for connection or communication is sent to a specific system.

Within a z/OS sysplex environment, VIPA is available. With VIPA, an IP address can be linked to the entire sysplex, and not to only one specific system inside this sysplex.

In order to do this, a port number in the sysplex environment must be reserved for the Hub Tivoli Enterprise Management Server. The Hub Tivoli Enterprise Management Server recognizes the sysplex VIPA as its host regardless of which system it is running on. This ensures that it is always reachable under the same IP address for all communication requests.

In order to simplify the configuration of the Tivoli components, z/OS internal communication uses only SNA. This avoids switching of ports between the Remote Tivoli Enterprise Management Server, its connected agents, and the Hub Tivoli Enterprise Management Server. The Tivoli Enterprise Management Server Hub must use its port exclusively across the entire z/OS sysplex.

If the Tivoli Enterprise Management Server Hub moves for any reason, the VTAM® APPLID moves with it. Thus, any components requesting communication with the Tivoli Enterprise Management Server Hub through SNA will find it.

Tivoli Enterprise Portal Server implementation based on Linux on IBM System z under z/VM

The Tivoli Enterprise Portal Server requires detailed attention. It runs only once in a single IBM Tivoli Monitoring V6 installation. Figure 1-16 shows the relevant components.

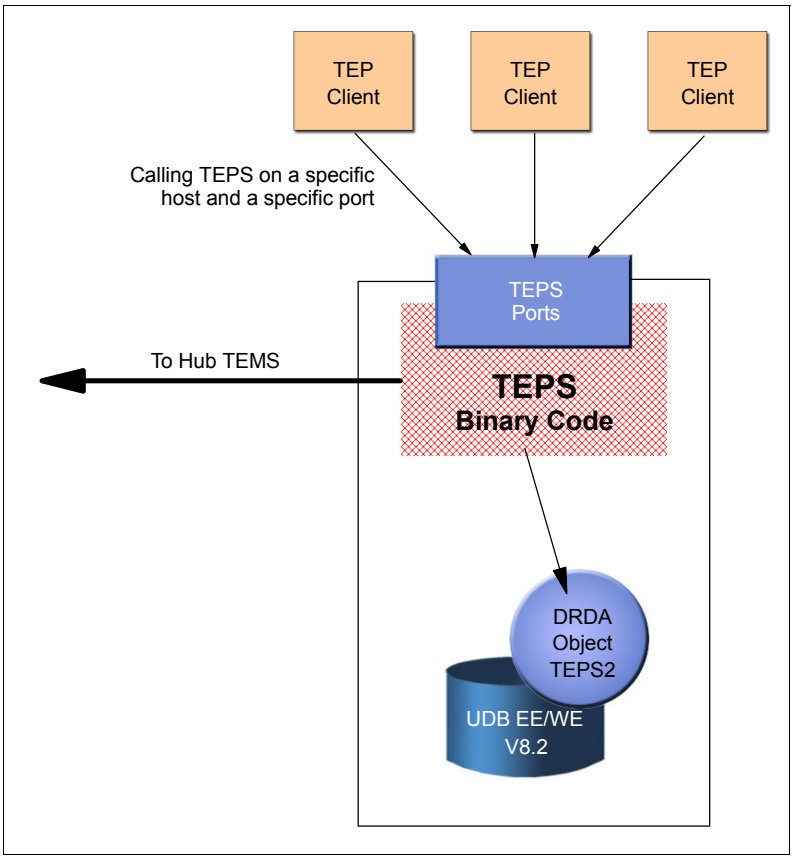


Figure 1-16 Simplified Tivoli Enterprise Portal Server component view

The Tivoli Enterprise Portal Server comprises the following three simple components:

- The Tivoli Enterprise Portal Server binary code, which implements the functions presented by the Tivoli Enterprise Portal Server:
  - It opens two ports for the Tivoli Enterprise Portal Server clients
  - It creates a connection to the Hub Tivoli Enterprise Management Server
  - It enables access to the underlying database system, using the Distributed Relational Database Architecture™ (DRDA®) Object
- The DRDA Object TEPS2

  This is the DB2 client part. It enables any program code to access a DB2 server installation, either locally or remotely. Because we use DB2 Universal Database, it is called DRDA, which is a common wording across Windows and Linux. On Windows, it is also called the Open Database Connectivity (ODBC) definition.
- DB2 Universal Database Workgroup Edition or Enterprise Edition.

  Within the database, all the presentation data, user settings, and so on, are stored. This database cannot be shared by multiple systems. The DB2 server must run locally, as stated in the installation manual.

These components must run on a single system. They cannot be distributed across multiple systems.

Using the Tivoli Enterprise Portal Client, the user calls a specific port on a defined host to connect to the Tivoli Enterprise Portal Server. If this host is not available for any reason, the user will not be able to access this service. A backup machine is hard to maintain because the database content has to be mirrored frequently to keep track of the changes taking place.

Virtualization with z/VM enables moving the entire system within seconds. The prerequisites for this are as follows:

- Shared discs across the z/VM hosting machines. Using z/VM, the user may leverage all the features from IBM System z.
  - Implemented high-availability features
    - Multiple data center
    - Disk mirroring across multiple locations
    - Nonstop power supply
    - Dynamic capacity adjustment
  - System z features
    - Security workload
    - Powerful network back plane
    - Resiliency

  The network is one of the features worth highlighting. The Tivoli Enterprise Portal Server has a huge network traffic. It serves all the Tivoli Enterprise Portal clients (the users) and it runs an extremely busy connection to the Hub Tivoli Enterprise Management Server. A powerful network connection is the key to useful performance for the users.
- Spare capacity in the backup z/VM hosting machine

The shared discs across the z/VM hosting machines feature is fulfilled within almost all the System z environments. For details about the spare capacity in the backup z/VM hosting machine, contact your local IBM Sales Representative.

1.4.5 Distributed systems and IBM System z working together

With this solution, a model that leverages the use of the Moveable Hub while keeping the Tivoli Enterprise Portal Server components on a platform used by most of the clients is available. Earlier, Tivoli Enterprise Portal Server was supported only on Windows platforms. Therefore, a lot of clients have their Tivoli Enterprise Portal Server running on Windows. Figure 1-17 shows such a solution approach when using the Moveable Hub on z/OS.

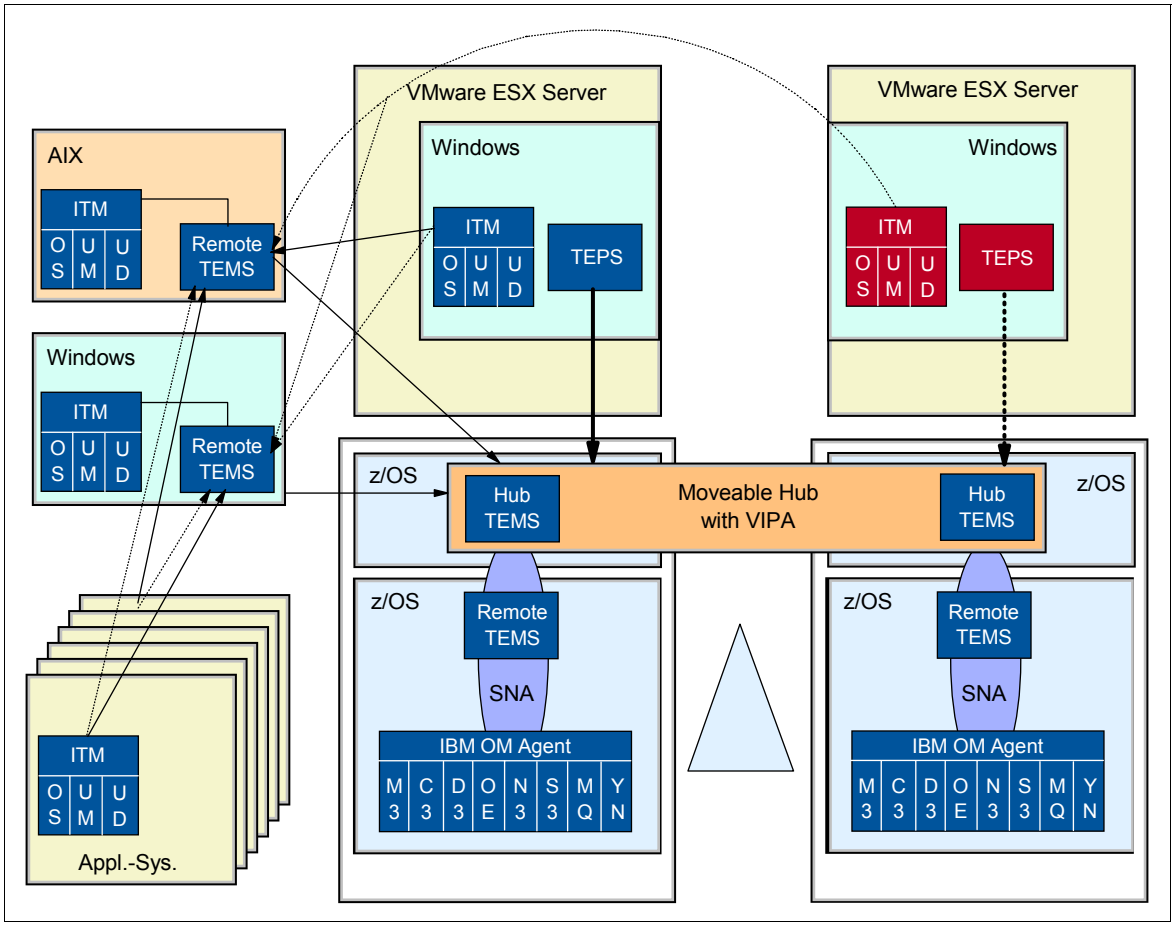


Figure 1-17 Windows Tivoli Enterprise Portal Server with Moveable Tivoli Enterprise Management Server Hub on z/OS

The components are:

- On Windows
  - Tivoli Enterprise Portal Server

    The Tivoli Enterprise Portal Server is placed on Windows. However, high availability functions and mirroring functions are not supported, and the switch in the Hot Standby Tivoli Enterprise Management Server is not recognized.
  - Remote Tivoli Enterprise Management Server

    The Windows-based Tivoli Enterprise Management Server Remote acts as the Primary Tivoli Enterprise Management Server for all Windows-based agents. It also hosts all the distribution packages for the Windows systems.

- IBM Tivoli Monitoring agents

  IBM Tivoli Monitoring agents for monitoring the server software, such as the operating system and the DB2 Universal Database, is installed on the Tivoli Enterprise Portal Server systems too. These agents connect to the Windows-based Tivoli Enterprise Management Server Remote as their primary server and to the AIX-based Tivoli Enterprise Management Server Remote as their secondary server. Thus, they have two ways in which to report to the Hub Tivoli Enterprise Management Server.

- On AIX
  - Remote Tivoli Enterprise Management Server

    The AIX-based Tivoli Enterprise Management Server Remote works as the primary Tivoli Enterprise Management Server for all AIX-based and Linux-based agents. It also hosts all the distribution packages for the AIX and Linux systems.

  - IBM Tivoli Monitoring agents

    The local agents (on this system, the operating system agent) connect only to the local Remote Tivoli Enterprise Management Server. If the machine is lost, which is one of the assumptions (1.4.2, “General architectural considerations” on page 32), a secondary connection will not help.

- On z/OS
  - Moveable Hub Tivoli Enterprise Management Server

    This feature is introduced in 1.4.4, “IBM System z environment scenario” on page 37. The idea is described in detail in “The Moveable Hub” on page 40. On z/OS, each LPAR hosts its own Remote Tivoli Enterprise Management Server. The communication with the Tivoli Enterprise Management Server Hub uses only SNA-based communication.

    The Tivoli Enterprise Management Server Hub is the only IBM Tivoli Monitoring V6 component on z/OS that supports TCP/IP communication.

  - On z/OS, each LPAR hosts its own Remote Tivoli Enterprise Management Server. The communication with the Tivoli Enterprise Management Server Hub uses only SNA-based communication.

  - All the z/OS-based agents connect to their local Tivoli Enterprise Management Server Remote by using only SNA. No other protocol is configured.

Because mirroring the Tivoli Enterprise Portal Server is an extremely time-consuming task that is difficult to maintain consistently, we offer a fourth solution for the clients who want to keep their Tivoli Enterprise Portal Server on systems based on IBM System x.

1.4.6 Virtualized IBM System x with Moveable Hub implementation

This architecture, as depicted in Figure 1-18, shows a way to get a high availability solution, where the Tivoli Enterprise Portal Server is located on IBM System x and the Tivoli Enterprise Management Server Hub resides on z/OS.

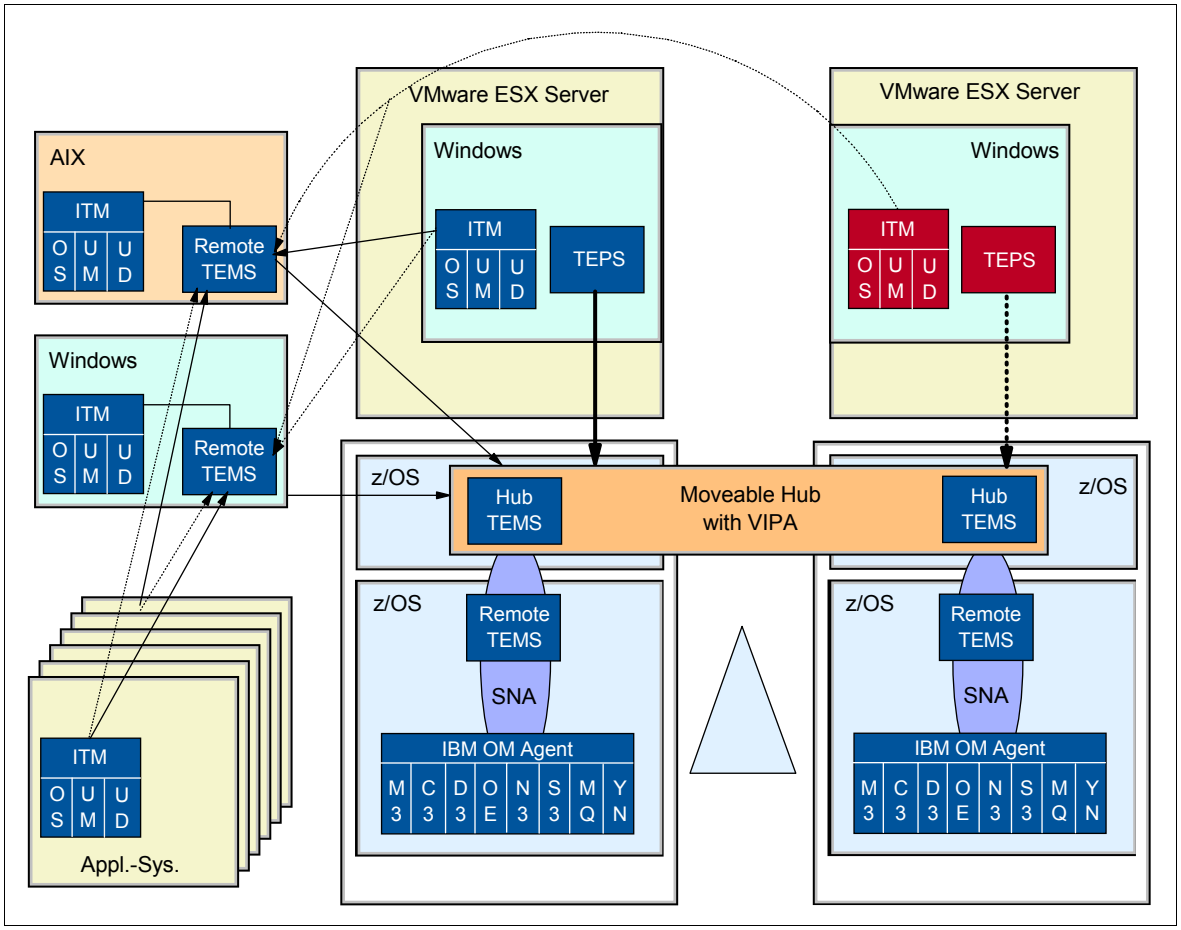


Figure 1-18 IBM System x-based virtualized Tivoli Enterprise Portal server and z/OS-based Moveable Hub

This solution is similar to that discussed in 1.4.4, “IBM System z environment scenario” on page 37. The Moveable Hub implementation is exactly the same. The only difference here is the virtualization platform under the Tivoli Enterprise Portal Server hardware. The components are:

- On VMware® ESX Server

  The VMware solution is discussed in detail in “VMware at a glance” on page 48.

  - On Windows
    - Tivoli Enterprise Portal Server

      The Tivoli Enterprise Portal Server is placed on Windows. It distributes the TCP/IP address of the virtual machine to the network. Even if the Tivoli Enterprise Portal Server moves, it retains this address.

    - On IBM Tivoli Monitoring agents

      IBM Tivoli Monitoring agents for monitoring the server software, such as the operating system and the DB2 Universal Database, are installed on the Tivoli Enterprise Portal Server systems too. These agents connect to the Windows-based

Tivoli Enterprise Management Server Remote as their primary server and to the AIX-based Tivoli Enterprise Management Server Remote as their secondary server. Thus, they have two ways in which to report to the Hub Tivoli Enterprise Management Server.

- On Linux

  VMware supports almost all the IBM System x-based operating systems, including Linux (SUSE® and Red Hat® distributions). For support information, contact your local IBM Sales Representative.

- On Windows
  - Remote Tivoli Enterprise Management Server

    The Windows-based Tivoli Enterprise Management Server Remote acts as the primary Tivoli Enterprise Management Server for all Windows-based agents. It also hosts all the distribution packages for the Windows systems.

    For Windows-based Remote Tivoli Enterprise Management Server, it is easy to add a secondary Tivoli Enterprise Management Server Hub statement.

  - IBM Tivoli Monitoring agents

    The local agents (on this system, the operating system agent) connect only to the local Remote Tivoli Enterprise Management Server. If the machine is lost, which is one of the assumptions (refer to 1.4.2, “General architectural considerations” on page 32), a secondary connection will not help.

- On AIX
  - Remote Tivoli Enterprise Management Server

    The AIX-based Tivoli Enterprise Management Server Remote functions as the primary Tivoli Enterprise Management Server for all AIX-based and Linux-based agents. It also hosts all the distribution packages for the AIX and Linux systems.

  - IBM Tivoli Monitoring agents

    The local agents (on this system, the operating system agent) connect only to the local Remote Tivoli Enterprise Management Server. If the machine is lost, which is one of the assumptions (refer to 1.4.2, “General architectural considerations” on page 32), a secondary connection will not help.

- On z/OS
  - Moveable Hub Tivoli Enterprise Management Server

    This feature is introduced in 1.4.4, “IBM System z environment scenario” on page 37. The idea is described in detail in “The Moveable Hub” on page 40. On z/OS, each LPAR hosts its own Remote Tivoli Enterprise Management Server. The communication with the Tivoli Enterprise Management Server Hub uses only SNA-based communication.

    The Tivoli Enterprise Management Server Hub is the only IBM Tivoli Monitoring V6 component on z/OS that supports TCP/IP communication.

  - On z/OS, each LPAR hosts its own Remote Tivoli Enterprise Management Server. The communication with the Tivoli Enterprise Management Server Hub uses only SNA-based communication.

  - All z/OS-based agents connect only to their local Remote Tivoli Enterprise Management Server, using SNA. No other protocol is configured.

VMware at a glance

As Figure 1-18 shows, the virtualization feature is now run by VMware that supports the IBM System x platform. The prerequisites for this solution are similar to that of z/VM:

- Shared disks across the VMware hosting machines
- Spare capacity in the backup VMware hosting machine

Figure 1-19 shows the principal configuration.

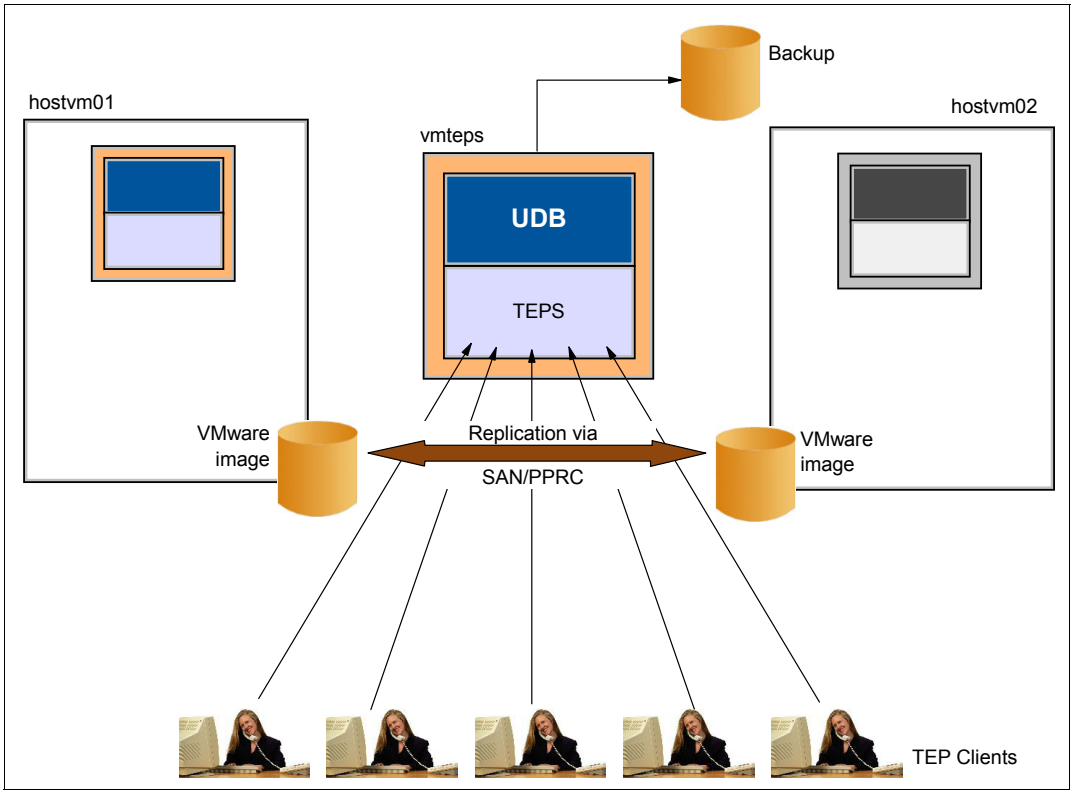


Figure 1-19 VMware setup for Tivoli Enterprise Portal Server system

The Tivoli Enterprise Portal Server is installed inside a virtual VMware. A configured machine inside VMware is represented by a VMware image. This image makes up a completely configured computer, including the following:

- Hardware definitions:
  - Memory
  - Hard disks
  - Network cards
  - Number of processors
- Software that has to be installed on it:
  - Operating System (Windows. However, it may also be Linux)
  - Database System (DB2 Universal Database (UDB) V8.2)
  - Tivoli Enterprise Portal Server software
- Run-time data:
  - Data stored in the database
  - State of the entire hosted system

By running a disk replication with Storage Area Network (SAN) or Peer-to-Peer Remote Copy (PPRC), the data of the VMware image is always ready for restart on a backup VMware host.

For more information about VMware, refer to the following Web site:

http://www.redbooks.ibm.com/redpapers/pdfs/redp3939.pdf

The result of virtualization is almost the same as that under z/VM on IBM System z.

The Tivoli Enterprise Portal Server can be executed on different hardware systems while retaining its known TCP/IP address. This is essential for the clients connecting to the portal server. Even in an emergency situation, the virtual Tivoli Enterprise Portal Server system can be restarted on another hardware system, assuming the VMware image is available.

This solution is effective if a VMware ESX Server implementation already exists. If not, contact your local IBM Sales Representative.

1.5 Conclusion

This chapter provided guidelines about the major points that must be kept in mind when picking the right alternative for your installation. Because many more combinations are possible than are documented in this book, we split the two major components, Tivoli Enterprise Portal Server and Tivoli Enterprise Management Server.

- Tivoli Enterprise Portal Server

  There are three ways of dealing with the missing backup and high availability mechanism:

  - Leave it as it is. However, this is not acceptable in many environments.
  - Create a standby solution as described in Chapter 12.3.5 of *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143. However, this option is fault intolerant with regard to the processes used to implement replication.
  - Virtualize the underlying machine and operating system by using z/VM or VMware

    This book focuses on this solution. By virtualizing the Tivoli Enterprise Portal Server hardware, the same network address can be retained when moving the machine from one location to another. In case of emergency, the Tivoli Enterprise Portal Server hardware may move to a backup location without changing any credentials for the user workstations.

- Tivoli Enterprise Management Server

  The Tivoli Enterprise Management Server Hub has been identified as the most important component inside the IT management infrastructure of IBM Tivoli Monitoring V6.

  Several methods exist to avoid the inherent problems:

  - Leave it as it is. Most of the time, however, this is not acceptable in production environments.
  - Implement the Hot Standby solution provided by the product. This is a good solution if only the distributed Tivoli Enterprise Management Server Remote is connected to the Hub Tivoli Enterprise Management Server. When a z/OS-based Tivoli Enterprise Management Server Remote cannot support a secondary Tivoli Enterprise Management Server statement in its setup, this solution is difficult to maintain if large z/OS-based environments are involved. Additionally, the Tivoli Enterprise Portal Server does not recognize the switch between the primary and the secondary Hub Tivoli Enterprise Management Server. Manual intervention is required.

– On z/OS, implement the Moveable Hub Tivoli Enterprise Management Server. This solution is useful for z/OS-dominant installations, where most of the managed systems are z/OS-based. If there is a large distributed environment, a performance degradation may take place due to excessive translation between American Standard Code for Information Interchange™ (ASCII) → Extended Binary Coded Decimal Interchange Code (EBCDIC) → ASCII. Refer to Chapter 5.1.3 in *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155, for more details.

– Virtualize the underlying machine and operating system by using z/VM or VMware. Because this solution has never been tested, this book does not discuss it. However, the advantages are clearly identifiable:

  • Network addresses never change
  • The Tivoli Enterprise Portal Server can automatically reconnect in any case
  • No overheads exist for the Hot Standby feature with regard to synchronization

  However, an IBM System x-based virtualization may suffer from network traffic problem. The virtual and physical network must support all the heartbeat signals and data requests the Tivoli Enterprise Management Server Hub handles. Over the years, z/VM has proved its capability to support high network traffic and CPU load.

The third and the fourth option have a common feature, that is, the IP address to the communication partners such as Tivoli Enterprise Management Server Remote and Tivoli Enterprise Portal Server will not change. This simplifies the configuration.

Although this book does not provide you with a final answer about where to place your central IBM Tivoli Monitoring V6 infrastructure components because this depends on various factors, it provides you with the necessary information that will help you make an informed choice.

2

IT environment of the Smart Bank showcase

This chapter describes the Smart Bank showcase operating mainframe environment used to implement the IBM Tivoli Monitoring service products, including the following:

- The Smart Bank showcase operating environment
- The hardware
- The software
- The workload

This chapter also describes how to configure the IBM Tivoli Monitoring service components with the Installation and Configuration Automation Tool (ICAT) for the following:

- IBM Tivoli Enterprise Monitoring Server
- IBM Tivoli Enterprise Monitoring Agent
- IBM Tivoli Monitoring Services
- IBM Tivoli OMEGAMON Classic Command and Menu Mode
- IBM Tivoli OMEGAMON II®

© Copyright IBM Corp. 2006. All rights reserved.

Figure 2-1 shows the overall technical architecture and how multichannel Automated Teller Machine (ATM), Teller, and Internet banking are implemented. This view describes the channel traffic across the different components.

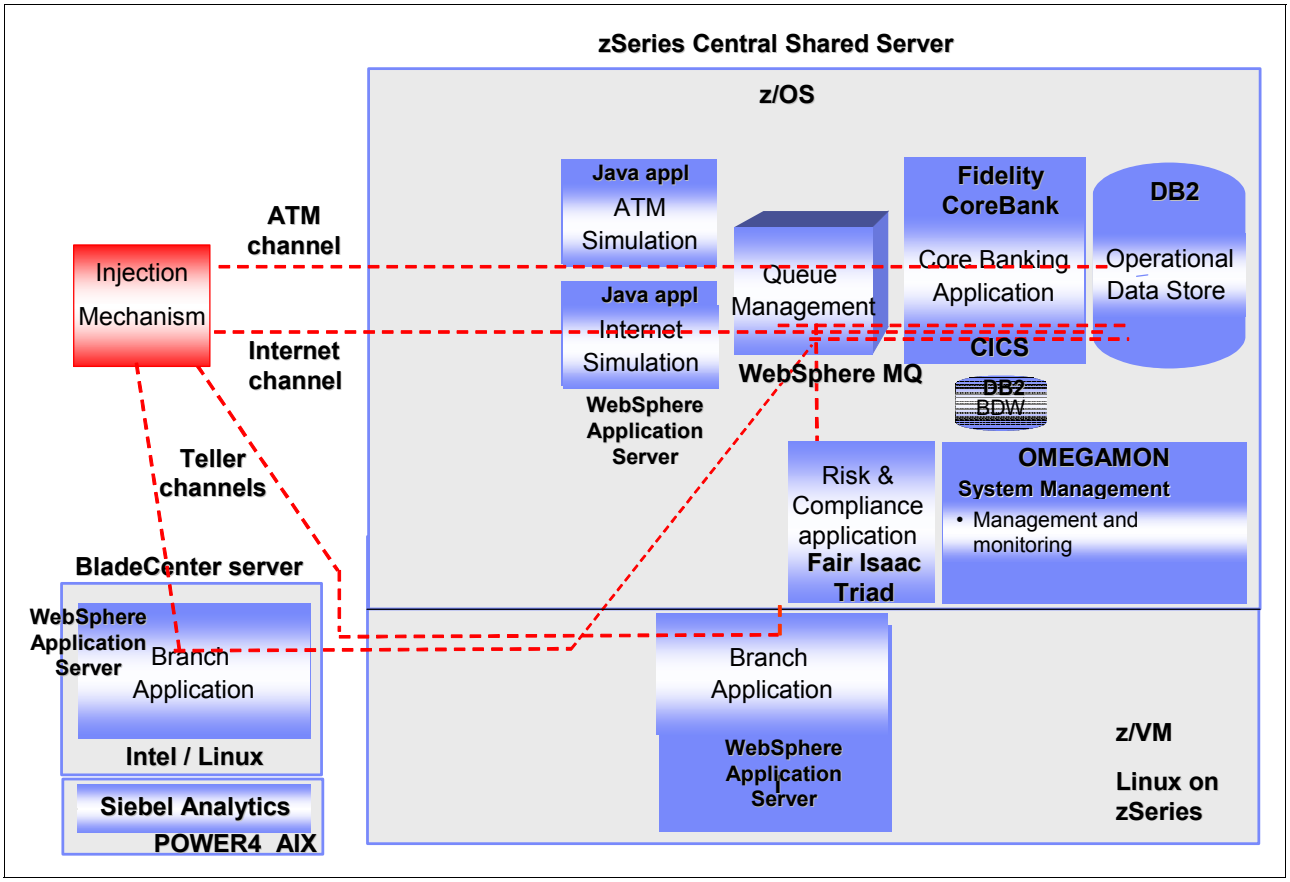


Figure 2-1 The Smart Bank's technical architecture

2.1 The IBM System z environment

This section describes the setup of the IBM System z environment for the Smart Bank demonstration. Integrated applications span WebSphere and Customer Information Control System (CICS) transactions with a high volume of data in a core database and a realistic scale of between 100 - 800 transactions per second. For high performance, workload management, infrastructure management, and high availability, the corebanking application and operating environment is based on z/OS in a Parallel Sysplex.

2.1.1 The hardware components

The Smart Bank backend infrastructure is based on a mainframe with IBM z/OS configured in Parallel Sysplex.

The hardware configuration comprises the following components:

- One IBM System z: one z990 (2094) Model B16
  - Sixteen shared central processors (CPs)
  - Two z/OS logical partitions (LPARs)
  - One z/VM LPAR
  - Six z/linux Guest
  - One coupling facility LPAR
  - Sixteen giga byte (GB) of memory
  - Four ESCON® channels
  - One dedicated gigabit (Gb) Ethernet Open Systems Adapter (OSA) card
  - One dedicated fast Ethernet card
- One Enterprise Storage (ESS 2105) direct access storage device (DASD) subsystem with 6.6 terabyte (TB)
- One IBM System x (X335) for simulating the workload injection
- One IBM BladeCenter (8627) with 14 blades for multichannel application simulation and Tivoli infrastructure management
  - HS20 with Windows XP
  - HS20 with SUSE Linux 8.0
  - JS20 POWER4 with AIX V5.2
- One CISCO 6509 switch

Figure 2-2 shows the hardware infrastructure setup for the Smart Bank test system that is used in this book.

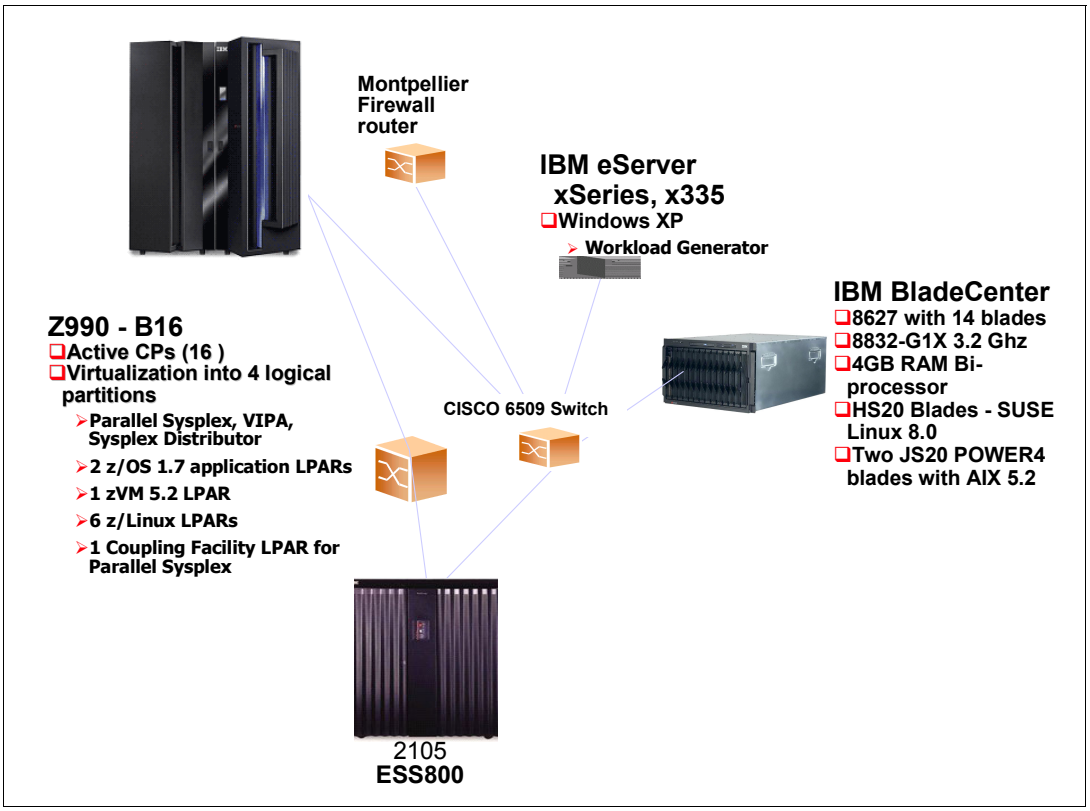


Figure 2-2 Hardware infrastructure components

The Smart Bank operating environment is not just a collection of product demonstrations. It is a real z/OS operating environment with real-life mixed workloads generated by real business applications. It is a large-scale, operational environment that simulates major application processes in a large bank. The database owns six million clients and 12 million client accounts and is running a complex transactional workload between 100 - 800 transactions per second. The corebanking application is based on an Independent Solution Vendor (ISV) (Fidelity) application named CoreBank V4.2.

2.1.2 The operating systems and the middleware components

The project is based on z/OS for the corebanking backend infrastructure. Following are the components:

- z/OS 1.7 in Parallel Sysplex mode with:
  - Two z/OS V1.7 LPARs, namely BA01 and BA02
  - Sysplex named BAPLEX
  - One CF LPAR named CF1
  - One LPAR with z/VM V5.2
  - Five Linux on IBM System z guests in SUSE V9 (64-bit usage)
  - One Linux on IBM System z guest in SUSE V9 (32-bit usage)
- Communication Server V1.7 as the network server, with Systems Network Architecture (SNA) and TCP/IP configured for both LPARs.

- This project uses a private giga bit network (as shown in Figure 2-3) for workload injection and for the Tivoli monitoring infrastructure. All the other remote hubs and agents in the distributed platform use the dynamic virtual IP address (DVIPA) address:
  - 10.1.1.20 for BA01
  - 10.1.1.21 for BA02
  - 10.1.1.29 for the DVIPA address

```
RO *ALL,D TCPIP,,NETSTAT,HOME
IEF196I IEF237I 782A ALLOCATED TO SYS00250
EZZ2500I NETSTAT CS V1R7 TCPIP 374
HOME ADDRESS LIST:
ADDRESS          LINK             FLG
10.1.1.20        GEOLINK
200.1.1.20       HECLINK          P
190.1.1.1        EZASAMEMVS
10.1.1.29        VIPL0A01011D     I
200.1.1.5        VIPLC8010105     I
190.1.1.1        EZAXCF02
127.0.0.1        LOOPBACK
8 OF 8 RECORDS DISPLAYED
END OF THE REPORT
EZZ2500I NETSTAT CS V1R7 TCPIP 185
HOME ADDRESS LIST:
ADDRESS          LINK             FLG
10.1.1.21        GEOLINK
200.1.1.21       HECLINK          P
190.1.1.2        EZASAMEMVS
190.1.1.2        EZAXCF01
10.1.1.29        VIPL0A01011D
200.1.1.5        VIPLC8010105
127.0.0.1        LOOPBACK
8 OF 8 RECORDS DISPLAYED
END OF THE REPORT
```

Figure 2-3 Smart Bank networking for the test system

- CICSPlex/SM V3.1 (CPSM) is used for CICSPlex Single System Image management, providing a single point of control and dynamic workload transaction balancing over the corebank CICSplex. CPSM is configured with:
  - Two coordinating address space (CAS), one per z/OS LPAR
  - Two CICSPlex/SM address space (CMAS), one per z/OS LPAR
- CICS Transaction Server V3.1 (CICS TS) is used as the backend corebanking COBOL application to host the CoreBank 4.2 application. It is configured with:
  - Two terminal-owning regions (TOR), one per z/OS LPAR
  - Six application-owning regions (AOR), three per z/OS LPAR
- IBM DB2 V8.1 is used for data store and is configured with two DB2 instances in data sharing mode:
  - One for operational corebanking application data store using the financial services data model (FSDM) with two members, one per z/OS LPAR
  - One for data warehouse using the model banking data warehouse model (from the Dublin software group [SWG]) with two members, one per z/OS LPAR

- WebSphere MQ for z/OS V5.3.1 is used as the classical access layer for corebanking and for publishing. It is configured with:
  - Two queue managers, one per z/OS LPAR
  - Two channel initiators, one per z/OS LPAR
- WebSphere Application Server V5.0.2 with WebSphere Business Integration Foundation Server V5.1 (WBIFS) is used for the ATM, Teller, and Internet banking channel applications and for the service-oriented architecture (SOA). It is configured with one servant region in BA01.
- WebSphere Application Server V6 is used for the ATM, Teller, and Internet banking channel application. It is configured with two servant regions, one per z/OS LPAR.
- CICS Transaction Gateway (CTG) V6 is used for J2EE connectors and for the SOA activity. It is configured with two CTG instances, one per z/OS LPAR.
- CICS Business Event Publisher (BEP) V1.2 is used to publish and populate the BDW database in real time. It is configured with:
  - Three address spaces in BA01, one data space server, one message server, and one DB2 event collector
  - A Batch for DB2 (BDW) population in BA01
- Named Counter Server is used by the corebanking COBOL application. It is configured with two address space servers, one per z/OS LPAR.

Figure 2-4 shows the operating systems and middleware infrastructure used for the Smart Bank development and for this book.

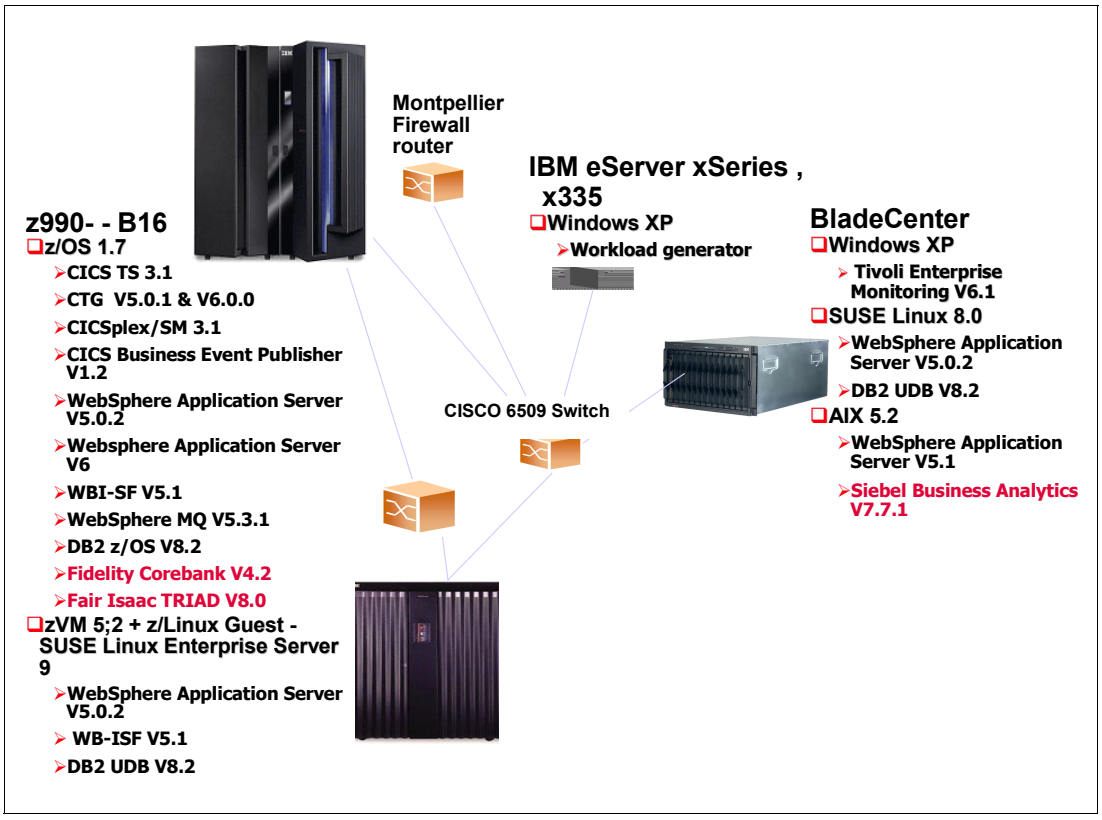


Figure 2-4 The operating systems and the middleware components

2.1.3 IBM Tivoli components

This demonstration simulates 24 hours in a bank operation through a realistic scenario. Many Tivoli System Management products and monitoring tools are used to track, monitor, and react to different events in a day. These include the following:

- IBM Tivoli Monitoring Services V6.1 (Product number 5698-A79) is the foundation on which all IBM Tivoli OMEGAMONs (mainframe and distributed) are built. This replaces the old CT/engine 350 with the CT/engine foundations. Following are the three components included in the IBM Tivoli Monitoring Services solution:
  - IBM Tivoli Enterprise Portal, earlier called CandleNet Portal
  - IBM Tivoli Enterprise Portal Server, earlier called CandleNet Portal Server
  - IBM Tivoli Enterprise Management Server
- IBM Tivoli OMEGAMON XE for CICS on z/OS V3.1.0 (Product number 5698-A58) provides a comprehensive means of gathering information to detect and prevent problems within the CICS regions.
- IBM Tivoli OMEGAMON XE on z/OS V3.1.0 (Product number 5698-A59) provides comprehensive performance information covering a single z/OS image or the sysplex level. Tivoli OMEGAMON XE on z/OS V3.1.0 integrates Tivoli OMEGAMON XE for sysplex, Tivoli OMEGAMON XE for OS/390®, and Tivoli OMEGAMON XE for cryptographic coprocessor.
- IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1.0 (Product number 5698-A46) provides a comprehensive monitor for z/OS I/O subsystem performance and storage availability. It also includes the IBM DS6000™ and IBM DS8000® device support.
- IBM Tivoli OMEGAMON XE for Mainframe Networks V3.1.0 (Product number 5698-A40) provides the ability to monitor and manage the health of the networks. It is used for TCP/IP, Virtual Telecommunications Access Method (VTAM), File Transfer Protocol (FTP), TN3270 sessions, and OSA.
- IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS (Product number 5655-P07) provides a tool to monitor, analyze, optimize, and tune the performance of the IBM Universal DataBase (UDB) and IBM DB2 on z/OS. It includes performance data and analysis tools for a performance warehouse.
- IBM Tivoli OMEGAMON XE for WebSphere Business Integration V1.1.0 (Product number 5698-A587) comprises several components:
  - Tivoli OMEGAMON XE for WebSphere Integration Brokers to monitor, verify, analyze, and tune message broker topologies
  - Tivoli OMEGAMON XE for WebSphere MQ Monitoring to monitor, collect, and analyze the WebSphere MQ-specific data on the queue manager
  - Tivoli OMEGAMON XE for WebSphere MQ Configuration to simplify the WebSphere MQ administration
- IBM Tivoli OMEGAMON XE for UNIX System Services (USS) V2.2.1 (Product number 5608-C15) is a required component to monitor the USS in OS/390 or z/OS.

- The IBM Tivoli Composite Application Manager products is a family of products that enable and provide the monitoring, tracking, and analysis of Java 2 Enterprise Edition (J2EE) applications. This technology allows the opening of the Java “Black Box” to see what is happening within the application. The IBM Tivoli Composite Application Manager consists of a common services address space (only for z/OS), a managing server and data collectors (for WebSphere Application Server, CICS, Information Management System [IMS], WebSphere MQ). The following components are used:
  - IBM Tivoli Composite Application Manager Common Services V6.00.00 (Product numbers: 5698-A69, 5698-A70, and 5698-A71) are IBM Tivoli Composite Application Manager components’ optional features that exist only in a z/OS environment where WebSphere Application Servers are monitored. It provides Workload Manager (WLM) information and serves as a repository for the System Management Facility (SMF) records generated. This repository takes copies of the SMF records and provides the basis for real-time analysis.
  - IBM Tivoli Composite Application Manager for SOA V6.0.0 (Product number 5698-A77) provides the function for managing the services and mediations in an SOA. It provides a simple control of message traffic between Web services.
  - IBM Tivoli Composite Application Manager for Response Time Tracking for z/OS V6.0.0 (Product number 5698-A75) is the new version of the IBM Tivoli Monitoring for Transaction Performance (TMTP) for z/OS. This product captures detailed performance data for all e-business transactions.
  - IBM Tivoli Composite Application Manager for CICS Transactions V6.00.00 (Product numbers 5698-A69 and 5698-A75) are the data collectors for CICS transactions. They help profile the “composite” transactions that originate in J2EE and branch off to CICS.
  - IBM Tivoli Composite Application Manager for WebSphere on z/OS V6.00.00 (Product number 5698-A71) is a data collector for WebSphere Application Server on z/OS. It allows J2EE administrators, application support analysts, or subject matter experts to profile “composite” transactions.
  - IBM Tivoli Business Systems Manager V3.1.0 (Product number 5698-A21) is an object-oriented systems management application that provides monitoring status and event management of the technical resource, application, and subsystems. It manages, collects, and correlates information within the z/OS environment from a variety of sources.

Note: IBM Tivoli Composite Application Manager for Response Time Tracking, IBM Tivoli Composite Application Manager for CICS transaction, IBM Tivoli Composite Application Manager for WebSphere on z/OS, and Tivoli Business System Manager were *not* used for this project.

Figure 2-5 shows the monitoring products and tools used in this project.

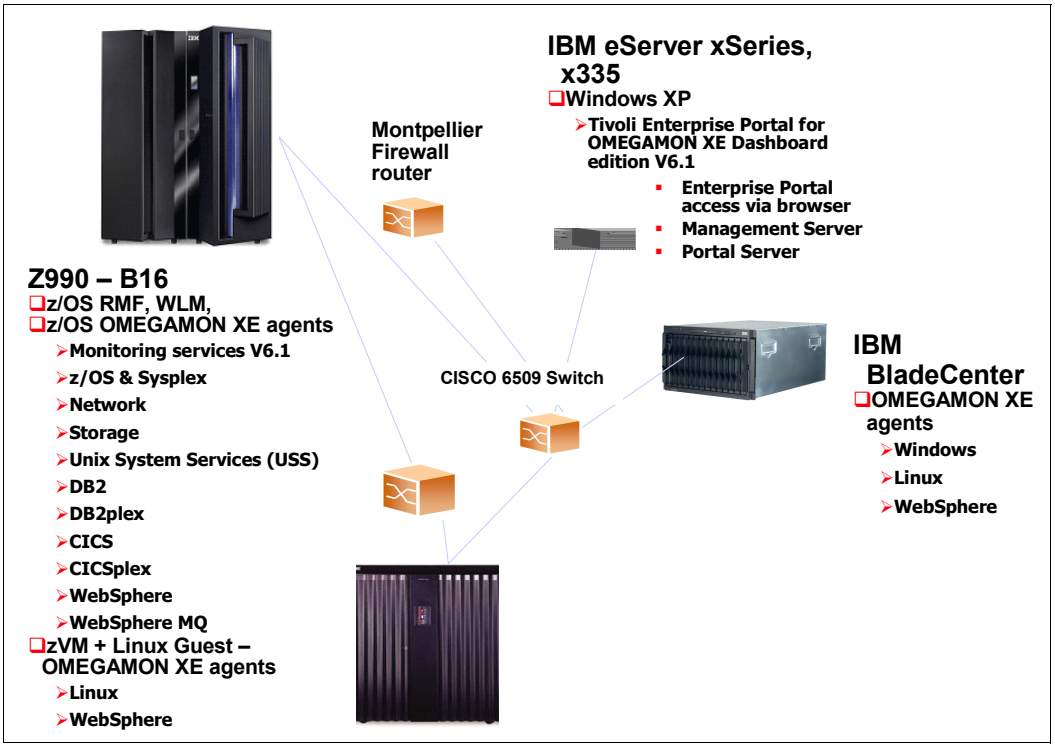


Figure 2-5 Tivoli system monitoring infrastructure

2.1.4 Summary

Figure 2-6 describes the complete backend corebanking infrastructure view for the Smart Bank project test environment used in this book.

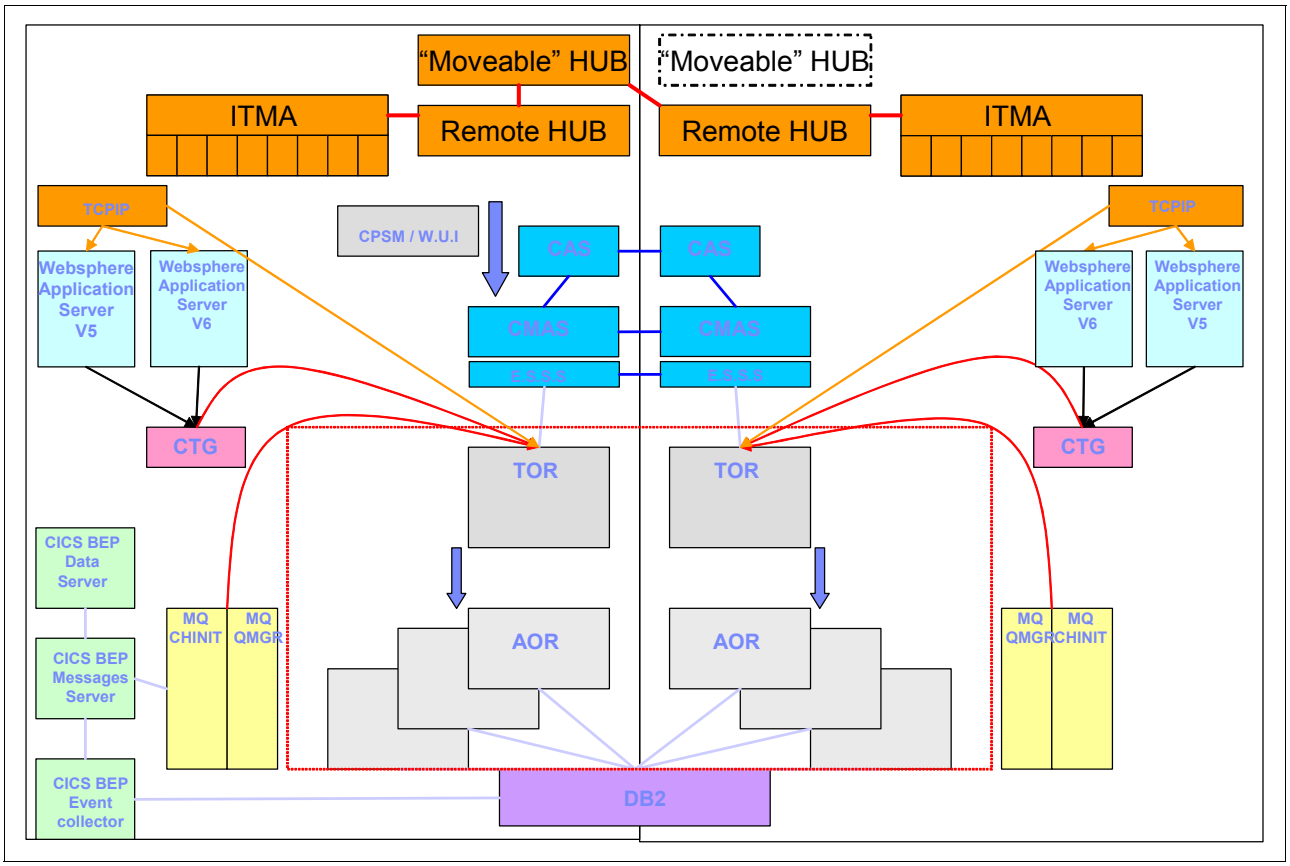


Figure 2-6 Detailed corebanking back office infrastructure view

2.1.5 Configuring the Smart Bank showcase

The installation and customization process can be split into the following multiple tasks:

- The System Modification Program/Extended (SMP/E) installation
- The use of the Installation and Configuration Assistant Tool (ICAT)
- The run-time environment building
- Configuring the primary hub
- Configuring the remote hub

The System Modification Program/Extended installation

The IBM Tivoli Monitoring Services V6.1 and Tivoli OMEGAMON XE for z/OS V3.1.0 products are delivered using the standard IBM enterprise software fulfillment processes either through electronic delivery or physical delivery. The installation is performed using the standard System Modification Program/Extended (SMP/E) process. This IBM Redbook does *not* discuss the SMP/E installation. For details about the complete SMP/E installation, refer to *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155.

The use of the Installation and Configuration Assistant Tool

The configuration of the IBM Tivoli Monitoring Services and Tivoli OMEGAMON XE products is made by an IBM configuration tool named ICAT. This tool allows easy configuration and assists you in defining your run-time environment. ICAT can be invoked either through the batch mode or the interactive mode. This book uses the interactive mode based on a set of Interactive System Productivity Facility (ISPF) dialogs to collect customization values and execute the creation of the run-time environment.

Important: We recommend that you make a note of the following points:

- For IBM Tivoli Composite Application Manager V6, only the IBM Tivoli Composite Application manager for SOA can be configured by ICAT
- Other IBM Tivoli Composite Application Manager components such as Response Time Tracking, client Information Control System (CICS) transaction, WebSphere, and Common Services cannot be configured through ICAT. Refer to the corresponding installation publications for details:
  - For IBM Tivoli Composite Application Manager for Response Time Tracking for z/OS:

    IBM Tivoli Composite Application Manager for Transaction Tracking Installation and Configuration Guide, GC32-9482
  - For IBM Tivoli Composite Application Manager for WebSphere z/OS:

    IBM Tivoli Composite Application Manager for WebSphere Installation and Customization Guide, GC32-9506
  - For IBM Tivoli Composite Application Manager for CICS transactions:

    IBM Tivoli Composite Application Manager for CICS Transactions Product Guide, SC32-9510
  - For IBM Tivoli Composite Application Manager Common Services:

    IBM Tivoli Composite Application Manager for WebSphere Operator's Guide, SC32-9508

ICAT performs the following tasks:

- Defines and builds a run-time environment that is used to run the product and plug the agent
- Configures the system parameters that are used for the product components
- Completes the configuration by setting up the required system definitions:
  - Creating the started task
  - Virtual Telecommunications Access Method (VTAM) definitions
  - Authorizing data sets, and so on

These processes must be performed individually for each of the monitoring agent product that is to be installed.

A full and complete description of the IBM ICAT is detailed in *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155.

Before using the ICAT, perform the following tasks:

1. After installing the contents of the product tape into your SMP/E environment, copy the contents of the KCIINST file into your INSTLIB in order to be able to start the configuration tool.
2. Start the configuration tool from an Interactive System Productivity Facility/Time Sharing Option (ISPF/TSO) session by invoking the newly created library:

   ```
   EXEC '(your.qual).INSTLIB'
   ```

   Note: The INSTLIB library must be created and members must be copied from the TKCIINST to the INSTLIB libraries earlier.

   After the ICAT copyright screen, the Main Menu (Figure 2-7) that allows you to set up the working environment is displayed:

   – Option 1 allows you to specify the allocation for the tool work libraries
   – Option 2 allows you to install the products and their maintenance. In our environment, we installed the SMP/E process instead of this option.
   – Option 3, Product to configure, allows you to customize each of the installed products to your environment. You can install only older OMEGAMON products using this option. All the new installations must be performed through the SMP process.
   – Option I allows you to provide the tool with the high-level qualifiers used in our environment (no defaults are given).
   – Option S, Services and Utilities, offers you different tools that are useful during the configuration. These include the Debug option, Modify CSI[1], Add new target in CSI, and so on.

```
---------------------------------- MAIN MENU ----------------------------------

 Enter the number to select an option:

    1   Set up work environment

    2   Install products or maintenance (for traditional Candle products only)

    3   Configure products

    I   Installation information
    S   Services and utilities

F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 2-7 ICAT Main Menu screen

Note: In the screens provided in the rest of this chapter, the last lines of the screen have been removed. The screens describe the keys you can use, and which do not provide any specific value to the displayed screens.

[1] Consolidated software inventory (CSI)

After you create and submit the allocate work libraries job, exit ICAT and allow the job to run before restarting the configuration tool.

Before configuring a product, set up the configuration environment by providing the high-level qualifier that you use in your environment, by selecting option 1, **Set up work environment**.

To configure a product, select option 3, **Configure products**, from the Main Menu (Figure 2-7). This displays the CONFIGURE PRODUCTS screen (Figure 2-8). Link the products to the run-time environment.

```
------------------------------ CONFIGURE PRODUCTS -----------------------------

 Enter the number to select an option:

    1   Set up configuration environment

    2   Select product to configure

    I   Configuration information
    S   Services and utilities
```

Figure 2-8 Configure Products screen

Building the run-time environment

The run-time environment is a set of z/OS runtime libraries and VSAM data sets required for the execution of Tivoli OMEGAMON XE products on a z/OS. There are three different types of run-time environment:

- FULL

  This is for allocating private and base libraries in a single set of data set. The FULL run-time environment is typically used for a single system image.

- BASE

  This is for allocating base libraries only. The BASE run-time environment is shared by all the z/OS participating images.

- SHARING

  This is for allocating private libraries only. Typically, you define a separate sharing run-time environment for each z/OS sysplex image that shares the base run-time environment. A sharing run-time environment must be used only in conjunction with either a base or a full run-time environment.

For each run-time environment and for each product you want to install, perform the following tasks:

1. Add a run-time environment

   This task (1) defines the run-time environment in the ICAT tool. Additional panels prompt for the required parameter values, which are in turn used to allocate the libraries and set the run-time environment default values for the products that are to be configured in this run-time environment. Adding a run-time environment creates a configuration tool definition only.

2. Build the libraries for the run-time environment

   It generates a batch job to allocate the required run-time libraries for the selected product. This job is presented for your review and submission. This task (2) must be performed for every run-time environment that contains the selected product. This includes the base run-time environments.

3. Configure the product in the run-time environment

   This task, which is not used for a base run-time environment, presents the panels to collect the parameter values required to configure the selected product for this run-time environment. The run-time environment default values are used when applicable. A batch job is generated and presented for your review and submission. This task (3) must be performed after step 2 for every run-time environment that contains the product.

4. Load the product into run-time environment

   This task loads all the product libraries after SMP/E. It generates a batch job to load the run-time libraries from the SMP/E target libraries for this run-time environment. Load the run-time environment libraries after SMP/E maintenance or product configuration (3).

   Note: In a sharing run-time environment type configuration, the master run-time environment and all the other run-time environments sharing the master run-time environment must be loaded.

5. Utilities display a selection menu to perform various useful processes for maintaining the run-time environment.

   This step is optional. However, one of this utilities is mandatory if you use the system variable for the hub.

You can build the run-time environment after configuring the product.

To configure a product, select option 2, **Select product to configure** in the screen shown in Figure 2-8. The Product Selection Menu screen (Figure 2-9) is displayed. This screen displays all the products that are installed and available from the installation in the system. In our example, Tivoli Monitoring Services on z/OS V6.1.0 is selected first. Depending on your environment, you may have a different list from which you can select a different product accordingly.

```
--------------------------- PRODUCT SELECTION MENU --------------------------

 Actions: S Select product

    IBM Tivoli Composite Application Manager for SOA V6.0.0
 S  IBM Tivoli Monitoring Services on z/OS V6.1.0
    IBM Tivoli OMEGAMON DE on z/OS V3.1.0
    IBM Tivoli OMEGAMON XE for CICS on z/OS V3.1.0
    IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS V3.1.0
    IBM Tivoli OMEGAMON XE for Mainframe Networks V3.1.0
    IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1.0
    IBM Tivoli OMEGAMON XE for WebSphere Business Integration V1.1.0
    IBM Tivoli OMEGAMON XE on z/OS V3.1.0
    OMEGAMON XE for OS/390 UNIX System Services V220
    OMEGAMON XE for WebSphere Application Server V130
F1=Help   F3=Back   F5=Refresh   F7=Up   F8=Down

COMMAND ===>
```

Figure 2-9 ICAT Product Selection Menu screen

After Tivoli Monitoring Services on z/OS V6.1.0 product is selected, a run-time environment screen is displayed (Figure 2-10). Because this is the first OMEGAMON product configured in this environment, no run-time environment is displayed. The run-time environment is not yet defined. Depending on how you want to use the run-time environment, you can either define a full run-time environment or a set of base run-time environment and sharing run-time environment.

```
------------------------ RUNTIME ENVIRONMENTS (RTEs) -------------------------

Actions: A Add RTE, B Build libraries, C Configure,
         L Load all product libraries after SMP/E,
         D Delete, U Update, V View values, Z Utilities

Action Name     Type    Sharing  Description

Enter=Next  F1=Help  F3=Back  F7=Up  F8=Down

COMMAND ===>
```

Figure 2-10 ICAT Run-time Environment screen

For the Smart Bank environment, we created one base run-time environment and three sharing run-time environments.

- One run-time environment type BASE named BASE
  - To allocate our base libraries. This environment is shared by all the participating images. This base run-time environment is used in conjunction with our sharing run-time environments.
  - To load all the products used by the sharing run-time environment
- One run-time environment type SHARING named BA01
  - To allocate our private libraries for the z/OS image (BA01)
  - To define and configure the Tivoli Enterprise Management Server Remote on BA01
  - To install and configure all Tivoli Monitoring for Applications on BA01
- One run-time environment type SHARING named BA02
  - To allocate our private libraries for the z/OS image (BA02)
  - To define and configure the Tivoli Enterprise Management Server Remote on BA02
  - To install and configure all Tivoli Monitoring for Applications on BA02
- One RTE type SHARING named HUB
  - To allocate our private libraries for the *moveable* HUB. It is not specific to one z/OS image.
  - To define and configure the primary HUB Tivoli Enterprise Management Server on BA01 or BA02

Note: No agents are installed and configured in this run-time environment type.

Figure 2-11 displays the run-time environments defined for the Smart Bank showcase.

```
------------------------ RUNTIME ENVIRONMENTS (RTEs) -------------------------

Actions: A Add RTE, B Build libraries, C Configure,
         L Load all product libraries after SMP/E,
         D Delete, U Update, V View values, Z Utilities

Action Name     Type    Sharing  Description

       BASE     BASE             RTE BASE for TIVOLI Smart Bank Sysplex BAPLEX
       BA01     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA01
       BA02     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA02
       HUB      SHARING BASE     RTE SHARE for moveable HUB Smart Bank

Enter=Next  F1=Help  F3=Back  F7=Up  F8=Down


COMMAND ===>
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 2-11 ICAT Run-time Environment screen for the Smart Bank environment

The shared run-time environments called BA01 and BA02, and the HUB run-time environment type are all shared with our base run-time environment type called BASE.

Working with z/OS system variables

When defining the run-time environment, it is possible to use the z/OS system variables as shown in Figure 2-12. By utilizing the system variables, the components inherit the system values from the z/OS system that they are started on.

```
 --------------------- UPDATE RUNTIME ENVIRONMENT (2 of 2)
---------------------

 Use z/OS system variables?     ==> Y (Y, N)
   RTE name specification       ==> &SYSNAME
   RTE base alias specification ==>
   Applid prefix specification  ==> K&SYSCLONE.
   Use VTAM model applids?      ==> N (Y, N)

 TEMS in this RTE      ==> Y   (Y, N)
   If Y, TEMS name     ==> BA01:TEMS                        (Case sensitive)

 VTAM communications values:
   Applid prefix       ==> CTD         Network ID     ==> PSSCBA
   Logmode table       ==> KDSMTAB1    LU6.2 logmode  ==> CANCTDCS

 TCP/IP communications values: (Required if TCP/IP is used in this RTE)
  *Hostname     ==> 10.1.1.20
  *Address      ==> 10.1.1.20
   Started task ==> TCPIP     (Recommended default = *)
   Port number  ==> 1918

  Enter=Next  F1=Help  F3=Back

 COMMAND ===>
```

Figure 2-12 ICAT: Run-time environment configuration screen

If the Use z/OS system variables? field is set to Y, a utility is necessary to create the system variables parameter member for the selected run-time environment. To create the system variables member, select **Z** in the Runtime Environment screen as shown in Figure 2-13.

```
 ------------------------ RUNTIME ENVIRONMENTS (RTEs)
------------------------

 Actions: A Add RTE, B Build libraries, C Configure,
          L Load all product libraries after SMP/E,
          D Delete, U Update, V View values, Z Utilities

 Action Name     Type    Sharing  Description

        BASE     BASE             RTE BASE for TIVOLI Smart Bank Sysplex BAPLEX
   Z    BA01     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA01
        BA02     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA02
        HUB      SHARING BASE     RTE SHARE for moveable HUB Smart Bank
 Enter=Next  F1=Help  F3=Back  F7=Up  F8=Down

 COMMAND ===>
```

Figure 2-13 ICAT: Run-time environment screen

A new screen (Figure 2-14) is displayed. Select **2** in the command area and then submit the generated job.

```
----------------------- RTE UTILITY MENU / RTE: BA01 ------------------------

 Specify the number of the desired utility.

   1  Create batch mode parameters
   2* Create System Variable parameter member
   3  Create System Variable VTAM major node rename job
   4  Create VTAM major node (one node for all products)
   5  Generate sample transport JCL
   6  Generate sample system procedure copy JCL
   7  Generate sample system VTAMLST copy JCL

 * Important: After the CB#VBA01 job runs, edit the RKANPAR(BA01)
              parameter member and follow the directions to ensure the
              proper resolution of cross-system variables.

F1=Help  F3=Back

OPTION ===> 2
```

Figure 2-14 ICAT: Run-time Environment Utility Menu

Configuring a Moveable Hub

The Tivoli Enterprise Monitoring Server Hub is the core part of Tivoli OMEGAMON XE platform. On one side, it connects with the Tivoli Enterprise Portal Server and on the other, it connects to the remote Tivoli Enterprise Monitoring Server. It is a backend for the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Monitoring Agent.

In our environment, we use the *Moveable* Tivoli Enterprise Monitoring Server Hub concept in z/OS Parallel Sysplex environment. This concept allows restarting a hub Tivoli Enterprise Monitoring Server on different members of the sysplex using a shared direct access storage device (DASD).

The Tivoli Enterprise Portal Server and Tivoli Enterprise Monitoring Server clients from other platforms (Windows, Linux, and AIX) are connected with the z/OS Moveable Tivoli Enterprise Monitoring Server Hub using TCP/IP. We used the Dynamic Virtual IP addressing (DVIPA function). The DVIPA can be retrieved by any member of the sysplex started by the Tivoli Enterprise Monitoring Server hub.

Tivoli Enterprise Monitoring Server in z/OS is connected to the moveable Tivoli Enterprise Monitoring Server Hub with Systems Network Architecture (SNA) communication.

Tivoli Enterprise Monitoring Agent in z/OS is connected to the Tivoli Enterprise Monitoring Server Remote in z/OS by an SNA communication.

Figure 2-15 shows the Tivoli monitoring infrastructure that is put in place for the Smart Bank showcase.

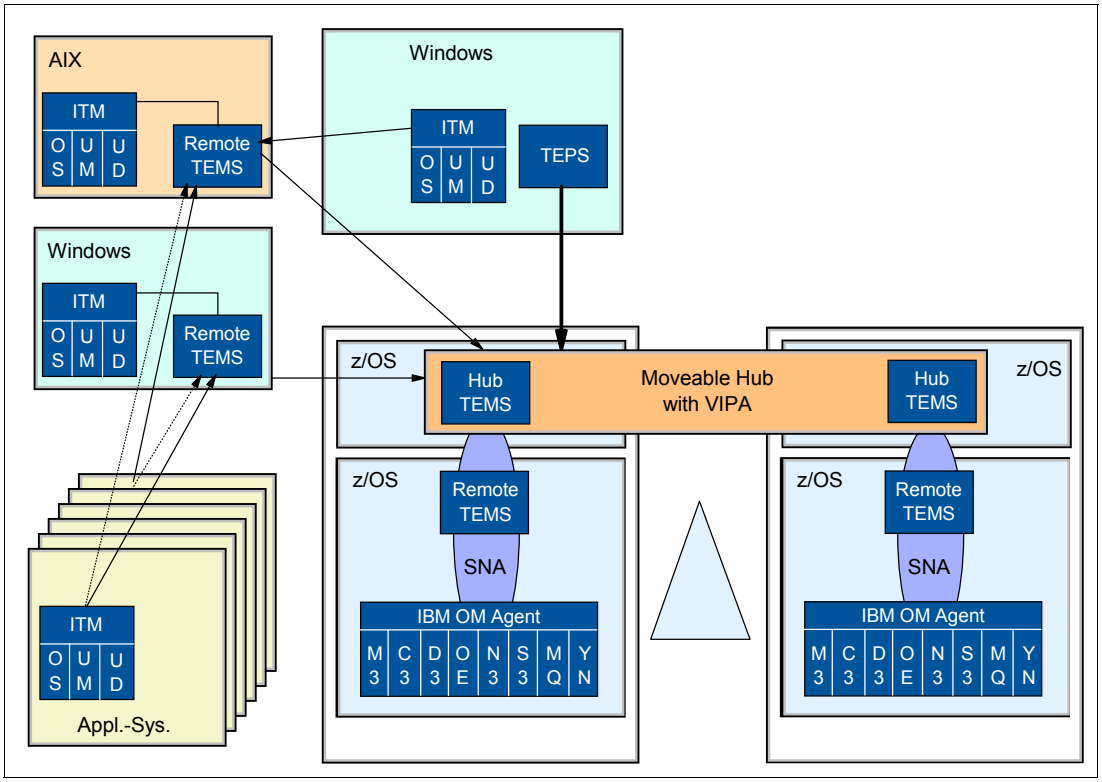


Figure 2-15 The Smart Bank Tivoli Monitoring Services infrastructure with the Moveable Hub

The configuration process

Following are the steps involved in configuring the primary Tivoli Enterprise Monitoring Server Hub in the z/OS sysplex environment:

1. In our case, Tivoli Monitoring Services on z/OS V6.1.0 is selected in the Product Selection screen (Figure 2-9 on page 65).
2. To configure Tivoli Enterprise Monitoring Server as the Tivoli Enterprise Monitoring Server Hub, select option **C**, as shown in Figure 2-16.

```
------------------------ RUNTIME ENVIRONMENTS (RTEs) ------------------------

Actions: A Add RTE, B Build libraries, C Configure,
         L Load all product libraries after SMP/E,
         D Delete, U Update, V View values, Z Utilities

Action Name     Type    Sharing  Description

       BASE     BASE             RTE BASE for TIVOLI Smart Bank Sysplex BAPLEX
       BA01     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA01
       BA02     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA02
  c    HUB      SHARING BASE     RTE SHARE for moveable HUB Smart Bank
Enter=Next  F1=Help  F3=Back  F7=Up  F8=Down
```

Figure 2-16 ICAT: Run-time Environment screen

This displays the screen shown in Figure 2-17. Use option **1** to configure the LU6.2 log mode, which generates a batch job you must submit.

```
------------ CONFIGURE THE TEMS (V610) / RTE: HUB ----------------------------

 Each RTE can contain only one TEMS.  To configure          Last selected
 the TEMS for this RTE, perform these steps in order:       Date     Time

   1  Create LU6.2 logmode                                 06/03/20  15:12
   2  Specify configuration values                         06/03/16  10:26
   3  Specify communication protocols                      06/03/20  15:19
   4  Create runtime members                               06/03/16  10:30
   5  Configure persistent data store
   6  Complete the configuration                           06/03/07  16:59

 Optional:

   7  View TEMS list and registration status               06/03/07  17:02
   8  Generate sample migration JCL

F1=Help  F3=Back
```

Figure 2-17 ICAT: Tivoli Enterprise Monitoring Server configuration screen

3. Select option **2** to specify the configuration values. Define the started task name, in this case, HUB, as shown in Figure 2-18. Set the security parameters for the integrated cryptographic services, if necessary. Select option **3** to specify the communication protocol.

```
------------------------ SPECIFY CONFIGURATION VALUES ------------------------

 To configure a TEMS for your site, complete the items on this panel.

   TEMS started task     ==> TIVSHUBT
   Hub or Remote?        ==> HUB

   Security settings:
     Security validation?  ==> N  (Y, N)
     Integrated Cryptographic Service Facility (ICSF) installed? ==> N (Y, N)
       ICSF load library
           ==> CSF.SCSFMOD0
       Encryption key
           ==> IBMTivoliMonitoringEncryptionKey


 Enter=Next  F1=Help  F3=Back  F5=Advanced

COMMAND ===>
```

Figure 2-18 ICAT: Tivoli Enterprise Monitoring Server Configuration Values screen

4. In the screen shown in Figure 2-19, specify the communication protocols you want the Tivoli Enterprise Monitoring Server to support. When performing this task, keep the following points in mind:
   - One protocol must be SNA
   - The communication protocols are used in the priority sequence you set

In the Smart Bank environment, we used SNA and TCP. After the communication protocol is set, press Enter.

```
---------------------- SPECIFY COMMUNICATION PROTOCOLS ----------------------

 Specify the communication protocols in priority sequence for
 TEMS HUB:CMS.

    IP.PIPE   ==> 2  (Non-secure NCS RPC)
    IP.UDP    ==>    (Non-secure NCS RPC)
    IP6.PIPE  ==>    (IP.PIPE for IPV6)
    IP6.UDP   ==>    (IP.UDP for IPV6)
    IP.SPIPE  ==>    (Secure IP.PIPE)
    IP6.SPIPE ==>    (Secure IP.PIPE for IPV6)
    SNA.PIPE  ==> 1  (Non-secure NCS RPC)

 Note:  One of the protocols chosen must be SNA.PIPE.
        * Web Services SOAP Server is enabled: TCP protocol is required.

Enter=Next  F1=Help  F3=Back

COMMAND ===>
```

Figure 2-19 ICAT: Communication Protocols screen

5. This displays the screen shown in Figure 2-20. Specify a unique IP address and a host name. The IP address used (in our example, 10.1.1.29) is a DVIPA. Within the z/OS sysplex distributor with VIPA, an IP address may be linked to the entire sysplex. It is not attributed to only one specific system image.

   This host name and the IP address are used by all the distributed Remote Hub and the agent to refer to the Hub and the Tivoli Enterprise Portal Server.

   On z/OS, each LPAR hosts its own Remote Tivoli Enterprise Monitoring Server. The communication with the Tivoli Enterprise Monitoring Server Hub uses SNA-based communication only. If the Tivoli Enterprise Monitoring Server Hub moves for any reason, the Virtual Telecommunications Access Method (VTAM) applid moves with it. Thus, any

component requesting communication with the Tivoli Enterprise Monitoring Server Hub through SNA will find it. This Hub implementation, called moveable Hub on z/OS, allows the restart of the Tivoli Enterprise Monitoring Server Hub on any member of the sysplex it is running on.

```
-------------------- SPECIFY IP.PIPE COMMUNICATION PROTOCOL -------------------

 Specify the IP.PIPE communication values for this TEMS.

 * Hostname                ==> 10.1.1.29
 * Address                 ==> 10.1.1.29
   Started task            ==> TCPIP     (Recommended default = *)

 Specify IP.PIPE and Web Services SOAP Server configuration.

   Port number             ==> 2018   (IP.PIPE)
   Port number             ==>        (IP.PIPE for IPV6)
   Port number             ==>        (Secure IP.PIPE)
   Port number             ==>        (Secure IP.PIPE for IPV6)
   HTTP server port number ==> 1921
   Access TEMS list vian SOAP Server?   ==> N (Y, N)
   Address translation     ==> N            (Y, N)
      Partition name       ==>

 * Note: See F1=Help for TSO HOMETEST command instructions.

 Enter=Next  F1=Help  F3=Back

COMMAND ===>
```

Figure 2-20 ICAT: IP.PIPE Communication Protocol configuration screen

6. In order to verify the SNA communication, display the VTAM applid for the Hub and the Remote Hub using the D NET command, as shown in Figure 2-21.

```
D NET,ID=TIVHUBLB,E
IST097I DISPLAY ACCEPTED
IST075I NAME = PSSCBA.TIVHUBLB, TYPE = APPL 236
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = CDSERVR
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST861I MODETAB=KDSMTAB1 USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=CANCTDCS USS LANGTAB=***NA***
.....
.....
IST171I ACTIVE SESSIONS = 0000000006, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I K01DSLB  ACTIV-S    CDD3C505D133D227 0A10 0001       PSSCBA
IST635I K02DSLB  ACTIV-S    CDD3C505D133D200 0A00 0001       PSSCBA
IST635I K01DSLB  ACTIV-P    CDD3C505D133D226 0001 0996       PSSCBA
IST635I K01DSLB  ACTIV/SV-P CDD3C505D133D225 0001 0001       PSSCBA
IST635I K02DSLB  ACTIV-P    CDD3C508D133E5D4 0001 09A0       PSSCBA
IST635I K02DSLB  ACTIV/SV-P CDD3C508D133E5D3 0001 0001       PSSCBA
IST314I END
```

Figure 2-21 VTAM applid display for the Hub

7. This screen (Figure 2-22) shows that the names of the sessions K01DSLB and K02DSLB are the active session names in our Hub Tivoli Enterprise Monitoring Server. Display the information for this session using another D NET command for the Remote Hub located in BA01. TIVHUBLB is the active session name for our Remote's Hub located in BA01. TIVHUBLB is the active session name of our Remote Hub in this example.

```
D NET,ID=K01DSLB,E
IST097I DISPLAY ACCEPTED
IST075I NAME = PSSCBA.K01DSLB, TYPE = APPL 243
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = CDSERVR
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST861I MODETAB=KDSMTAB1 USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=CANCTDCS USS LANGTAB=***NA***
....
....
IST171I ACTIVE SESSIONS = 0000000003, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I TIVHUBLB ACTIV-S    CDD3C505D133D226 09A3 0001       PSSCBA
IST635I TIVHUBLB ACTIV/SV-S CDD3C505D133D225 0001 0001       PSSCBA
IST635I TIVHUBLB ACTIV-P    CDD3C505D133D227 0001 0A1D       PSSCBA
```

Figure 2-22 VTAM applid display for the Remote Hub in LPAR BA01

All the z/OS-based agents connect to their local Tivoli Enterprise Monitoring Server Remote using only SNA. No other protocol is configured. Figure 2-23 shows the display provided after one of our Remote Hubs has started all the z/OS Tivoli Enterprise Monitoring Agents.

```
D NET,ID=KO1DSLB,E
IST097I DISPLAY ACCEPTED
IST075I NAME = PSSCBA.KO1DSLB, TYPE = APPL 136
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = CDSERVR
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST861I MODETAB=KDSMTAB1 USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=CANCTDCS USS LANGTAB=***NA***
IST171I ACTIVE SESSIONS = 0000000021, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I KO1WWNC  ACTIV-S    CDD3C505D133D2B5 016B 0001       PSSCBA
IST635I KO1OENC  ACTIV-S    CDD3C505D133D2B2 0061 0001       PSSCBA
IST635I KO1MCNC  ACTIV-S    CDD3C505D133D2AF 0351 0001       PSSCBA
IST635I KO1N3NC  ACTIV-S    CDD3C505D133D2AC 0061 0001       PSSCBA
IST635I KO1MVNC  ACTIV-S    CDD3C505D133D2A6 00B2 0001       PSSCBA
IST635I KO1M2EQ  ACTIV-S    CDD3C505D133D2A2 0022 0001       PSSCBA
IST635I TIVHUBLB ACTIV-S    CDD3C505D133D29E 1CE6 0001       PSSCBA
IST635I TIVHUBLB ACTIV/SV-S CDD3C505D133D29D 0001 0001       PSSCBA
IST635I KO1WWNC  ACTIV-P    CDD3C505D133D2B4 0001 09AF       PSSCBA
IST635I KO1WWNC  ACTIV/SV-P CDD3C505D133D2B3 0001 0001       PSSCBA
IST635I KO1OENC  ACTIV-P    CDD3C505D133D2B1 0001 0152       PSSCBA
IST635I KO1OENC  ACTIV/SV-P CDD3C505D133D2B0 0001 0001       PSSCBA
IST635I KO1MCNC  ACTIV-P    CDD3C505D133D2AE 0001 0A6D       PSSCBA
IST635I KO1MCNC  ACTIV/SV-P CDD3C505D133D2AD 0001 0001       PSSCBA
IST635I KO1N3NC  ACTIV-P    CDD3C505D133D2AB 0001 0045       PSSCBA
IST635I KO1N3NC  ACTIV/SV-P CDD3C505D133D2AA 0001 0001       PSSCBA
IST635I KO1MVNC  ACTIV-P    CDD3C505D133D2A5 0001 0081       PSSCBA
IST635I KO1MVNC  ACTIV/SV-P CDD3C505D133D2A4 0001 0001       PSSCBA
IST635I KO1M2EQ  ACTIV-P    CDD3C505D133D2A1 0001 0023       PSSCBA
IST635I KO1M2EQ  ACTIV/SV-P CDD3C505D133D2A0 0001 0001       PSSCBA
IST635I TIVHUBLB ACTIV-P    CDD3C505D133D29F 0001 15A5       PSSCBA
IST314I END
```

Figure 2-23 VTAM applid display for the Remote HUB in BA01 with all the z/OS agents started

8. In Figure 2-20 on page 73, when the IP communication protocol is set, press Enter. The SNA Communication Protocol screen (Figure 2-24) is displayed. Specify the SNA parameters. The applid prefix value is used to create all the VTAM APPLIDs required by the Tivoli Enterprise Monitoring Server. These applids begin with a prefix and end with a specific value that makes each applid unique. These applids are contained in a VTAM Major node.

```
---------------------- SPECIFY SNA COMMUNICATION PROTOCOL --------------------

 Specify the SNA communication values for this TEMS.

   Applid prefix          ==> TIVHUB
   Network ID             ==> PSSCBA   (NETID value from SYS1.VTAMLST(ATCSTRnn))


Enter=Next  F1=Help  F3=Back  F6=Applids

COMMAND ===>
```

Figure 2-24 ICAT: SNA Communication Protocol configuration screen

After the SNA communication protocol is set, press Enter. This takes you to the Tivoli Enterprise Monitoring Server configuration main menu shown in Figure 2-17 on page 70, from where the configuration process continues. In this screen, perform the following tasks:

a. Select option **4** to create the runtime members. These members are created in the runtime libraries for this run-time environment.

b. Select option **5** to configure the persistent data store. This step is optional. However, this option is required if you are using the data warehousing function.

c. Finally, select option **6** to complete the configuration. This generates a list of procedures that must be performed after you exit ICAT in order to finalize the installation and configuration of this product.

Configuring a Remote Hub on z/OS

This section describes how to configure a Remote Tivoli Enterprise Monitoring Server Hub in a z/OS sysplex environment.

In our example, Tivoli Monitoring Services on z/OS V6.1.0 is selected as the product to configure (see Figure 2-9 on page 65, with S selected in front of the product). This Tivoli Enterprise Monitoring Server is to be configured as the Remote Tivoli Enterprise Monitoring Server Hub.

Note: Only the process involved for one Remote Hub is described here. In our environment, we followed the same process for both the Remote Hubs Tivoli Enterprise Monitoring Server (BA01 and BA02).

Perform the following tasks:

1. Select action **C** to configure BA01 as the Remote Tivoli Enterprise Monitoring Server Hub, as shown in Figure 2-25.

> **Note:** From a terminology point, there is only one hub in a Tivoli Enterprise environment, the Primary hub. The Remote Tivoli Enterprise Monitoring Server Hub is now referred to as Tivoli Enterprise Monitoring Server Remote.

```
 ------------------------ RUNTIME ENVIRONMENTS (RTEs)
------------------------

 Actions: A Add RTE, B Build libraries, C Configure,
          L Load all product libraries after SMP/E,
          D Delete, U Update, V View values, Z Utilities

 Action Name     Type    Sharing  Description

        BASE     BASE             RTE BASE for TIVOLI Smart Bank Sysplex BAPLEX
   c    BA01     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA01
        BA02     SHARING BASE     RTE SHARE for TIVOLI Smart Bank LPAR BA02
        HUB      SHARING BASE     RTE SHARE for moveable HUB Smart Bank
 Enter=Next  F1=Help  F3=Back  F7=Up  F8=Down

 COMMAND ===>
```

Figure 2-25 ICAT: Run-time Environment screen

2. A screen (Figure 2-26) is displayed. Select option **1** to configure the LU6.2 logmode. Then, submit the generated job. Select option **2** to specify the configuration values.

```
------------ CONFIGURE THE TEMS (V610) / RTE: BA01 --------------------------

 Each RTE can contain only one TEMS.  To configure           Last selected
 the TEMS for this RTE, perform these steps in order:        Date     Time

   1  Create LU6.2 logmode                                  06/03/13  12:05
   2  Specify configuration values                          06/03/15  09:42
   3  Specify communication protocols                       06/03/15  09:42
   4  Create runtime members                                06/03/08  10:52
   5  Configure persistent data store                        06/03/03  10:37
   6  Complete the configuration                            06/02/27  14:00

 Optional:

   7  View TEMS list and registration status                06/03/01  15:57
   8  Generate sample migration JCL


OPTION ===>
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 2-26 ICAT: Tivoli Enterprise MonitoringServer configuration screen

3. In the Specify Configuration Values screen (Figure 2-27), define the Hub as `REMOTE` and set the security and integrated cryptographic service facility to `No`.

```
 ------------------------ SPECIFY CONFIGURATION VALUES
------------------------

  To configure a TEMS for your site, complete the items on this panel.

    TEMS started task     ==> TIVSREMT
    Hub or Remote?        ==> REMOTE   (Connected to ==> HUB:CMS)

    Security settings:
      Security validation?  ==> N  (Y, N)
      Integrated Cryptographic Service Facility (ICSF) installed? ==> N (Y, N)
        ICSF load library
            ==> CSF.SCSFMOD0
        Encryption key
            ==> IBMTivoliMonitoringEncryptionKey

Enter=Next  F1=Help  F3=Back  F5=Advanced

 COMMAND ===>
```

Figure 2-27 ICAT: Tivoli Enterprise Monitoring Server Configuration Values screen

4. A Tivoli Enterprise Monitoring Server Remote requires a Tivoli Enterprise Monitoring Server Hub connection. To select the Hub (Primary Tivoli Enterprise Monitoring Server) to connect to the Remote Tivoli Enterprise Monitoring Server, perform the following tasks:

   a. Select **F5** and **F10** to view the Tivoli Enterprise Monitoring Server list-specific panel shown in Figure 2-28 on page 79. The Tivoli Enterprise Monitoring Server eligible list that you have already configured is displayed. At this point, the importance of first defining the Tivoli Enterprise Monitoring Server Hub and then defining the Remote Tivoli Enterprise Monitoring Server, becomes clear. Select the Tivoli Enterprise Monitoring Server Hub that you want from the list and select **F5** to clear out any IP information.

```
----------------- COMMUNICATION SELECTION PANEL FOR BA01       Row 1 to 1 of 1
COMMAND ===>

 Select the hub (primary TEMS) to connect to this remote TEMS.
 The currently connected hub is:  HUB:CMS

 The following lists eligible Tivoli Enterprise Monitoring Servers
 (TEMS) that you have configured.  Enter S next to the TEMS to connect
 to this remote.  To manually enter the TEMS information, press F5.

    TEMS     RTE name     RTE
    Type    (mid-level)   Description
--------------------------------------------------------------------------------
 _  HUB      HUB          RTE SHARE for moveable HUB Smart Bank
 F1=Help  F3=Back  F5=Advanced  F7=Up  F8=Down


 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=          F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 2-28 ICAT: Tivoli Enterprise Monitoring Server eligible list

b. This displays the advanced communication screen shown in Figure 2-29. In this communication configuration screen, provide the primary HUB Tivoli Enterprise Monitoring Server name, the Global location applid, and the network. The Tivoli Enterprise Monitoring Server Remote uses this information to connect to the primary Hub Tivoli Enterprise Monitoring Server.

```
 ------------ SPECIFY CONFIGURATION - HUB VALUES FOR REMOTE TEMS
---------------

  You are about to configure a Remote TEMS.  Enter the following information
  about the Hub TEMS that the Remote TEMS connects to:
    Hub TEMS name (Case sensitive)       ==> HUB:CMS
    Global location broker applid of Hub ==> TIVHUBLB
    Network ID of Hub                    ==> PSSCBA

  If the Remote TEMS requires TCP support, enter the Hub's values for:
    Hostname of Hub        ==>
    Address of Hub         ==>

    If the Remote TEMS requires IP.PIPE and/or IP.UDP support, enter
    the Hub's values (if applicable):
       IP.PIPE port number   ==>       (Non-secure NCS RPC)
       IP6.PIPE port number  ==>       (IP.PIPE for IPV6)
       IP.SPIPE port number  ==>       (Secure IP.PIPE)
       IP6.SPIPE port number ==>       (Secure IP.PIPE for IPV6)
       IP.UDP port number    ==>       (Non-secure NCS RPC)
       IP6.UDP port number   ==>       (IP.UDP for IPV6)

  Enter=Next  F1=Help  F3=Back

 COMMAND ===>
```

Figure 2-29 ICAT: Advanced communication configuration screen

5. After the communication is made between the primary Hub and the Remote Tivoli Enterprise Monitoring Server, return to the main screen (Figure 2-26) and continue with the configuration options.

To verify the SNA communication between the primary Hub Tivoli Enterprise Monitoring Server, the remote Tivoli Enterprise Monitoring Server, and the agents, use a VTAM display applid as explained earlier, and displayed in Figure 2-23 on page 75.

The various agents that must be installed must also be configured. This is described in the individual installation and customization documentation.

2.1.6 OMEGAMON XE Tivoli enterprise management: Initialization checklist for the server and the agent

The OMEGAMON Engine Service is a collection of basic z/OS and communication service routines built specifically for the z/OS environment. All OMEGAMON XE Tivoli enterprise management server and Tivoli Enterprise management Agent address spaces that have been started, load and use the services of OMEGAMON Engine Service. The successful initialization is noted by the message KLVIN408 in the RKVLOG as shown in Example 2-1.

Example 2-1 IBM OMEGAMON Engine Service initialization

```
KLVIN408 IBM OMEGAMON PLATFORM ENGINE VERSION 400 READY ON BA01(0123392084):
```

OMEGAMON address spaces must use SNA either exclusively or in conjunction with TCP/IP. In the Smart Bank environment, we use SNA in conjunction with TCP/IP. This section discusses how to check the status of the communication protocol between the following:

- Between the z/OS Tivoli Enterprise Monitoring Server Hub and the z/OS Tivoli Enterprise Monitoring Server Remote for the connectivity to the Tivoli Enterprise Monitoring Server Hub and the Server Authentication callback
- Between the z/OS Tivoli Enterprise Monitoring Server Remote and the z/OS Tivoli Enterprise Monitoring Agent
- Between the z/OS Tivoli Enterprise Monitoring Server Hub and the Tivoli Enterprise Portal Server on Windows

SNA initialization between z/OS Tivoli Enterprise Monitoring Server Hub and z/OS Tivoli Enterprise Monitoring Server Remote

For the z/OS Tivoli Enterprise Monitoring Server Hub on BA01, a successful SNA initialization message is registered in the RKVLOG, as shown in Example 2-2.

Example 2-2 z/OS Tivoli Enterprise ManagementServer Hub SNA initialization message on BA01

```
BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="TIVHUBLB"
BSS1_GetEnv") KDCFP_TPNAME=KDCFC_TPNAME=KLXBS_TPNAME="SNASOCKETS"
BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
"getEnv") AF_SNA configuration: Alias(TIVHUBLB) Mode(CANCTDCS) TpName(SNASOCKETS)
....
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe
```

For the z/OS Tivoli Enterprise Monitoring Server Remote on BA01, a successful SNA initialization message is registered in the RKVLOG, as shown in Example 2-3.

Example 2-3 z/OS Tivoli Enterprise Monitoring Server Remote SNA initialization message on BA01

```
"BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="KO1DSLB"
"BSS1_GetEnv") KDCFP_TPNAME=KDCFC_TPNAME=KLXBS_TPNAME="SNASOCKETS"
"BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
,"getEnv") AF_SNA configuration: Alias(KO1DSLB) Mode(CANCTDCS) TpName(SNASOCKETS)
.......
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe
```

For the z/OS Tivoli Enterprise Monitoring Server Remote on BA02, a successful SNA initialization message is registered in the RKVLOG, as shown in Example 2-4.

Example 2-4 z/OS Tivoli Enterprise MonitoringServer Remote SNA initialization message on BA02

```
"BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="KO2DSLB"
"BSS1_GetEnv") KDCFP_TPNAME=KDCFC_TPNAME=KLXBS_TPNAME="SNASOCKETS"
"BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
,"getEnv") AF_SNA configuration: Alias(KO2DSLB) Mode(CANCTDCS) TpName(SNASOCKETS)
........
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe
```

In these examples:

- *KDCFC_ALIAS* identifies the APPL definition of the independent Logical Unit (LU) to be used in this process.
- *KDCFC_MODE* identifies the LOGMODE name used. The same name is found in the LOGMODE specification of the KDCFC_Alias APPL definition.
- *KDCFC_TPNAME* is the Transaction Processing Name.
- *KDE1I_OpenTransportProvider* is the last message that indicates that the SNA transport message is operational.

Many OMEGAMON processes build and query a list of possible Tivoli Enterprise Monitoring Server addresses called *server list*. This server list contains Local Location Broker (LLB) and Global Location Broker (GLB).

- The LLB entries of the server list are derived from the GLB.
- The GLB entries of the server list are built from the content of the KDCSSITE member of RKANPAR.

Example 2-5 displays the server list for the Tivoli Enterprise Monitoring Server Hub.

Example 2-5 Server list for a Tivoli Enterprise Monitoring Server Hub

```
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe (008B)
"KDE1I_OpenTransportProvider") Transport opened: socket/ip.pipe (00A2)
"NewSDB") LLB entry 1 is sna:(PSSCBA.TIVHUBLB.CANCTDCS.SNASOCKETS).135.
"NewSDB") LLB entry 2 is ip.pipe:£10.1.1.29.2018.
"NewSDB") GLB entry 1 is sna:(PSSCBA.TIVHUBLB.CANCTDCS.SNASOCKETS).135.
"NewSDB") GLB entry 2 is ip.pipe:£10.1.1.29.2018.
"NewSDB") GLB entry 3 is sna:(PSSCBA.TIVHUBLB.CANCTDCS.SNASOCKETS).135.
"NewSDB") GLB entry 4 is ip.pipe:£10.1.1.29.2018.
KDSNC004   Bind of local location broker complete= sna:
(PSSCBA.TIVHUBLB.CANCTDCS.SNASOCKETS)
KDSNC004   Bind of local location broker complete= ip.pipe
:£10.1.1.29.2018.
KDSNC007   Local Location Broker is active
KDSNC008   Global Location Broker is activ
```

SNA initialization between z/OS Tivoli Enterprise Monitoring Server Remote and z/OS Tivoli Enterprise Monitoring Server

For the z/OS Tivoli Enterprise Monitoring Agent, a successful SNA initialization message is registered in the RKVLOG for BA01 and BA02. The example shown in Example 2-6 is based on Tivoli OMEGAMON II for Multiple Virtual Storage (MVS) (Tivoli Enterprise Monitoring Server) RKVLOG messages.

Example 2-6 z/OS Tivoli OMEGAMON XE for MVS on z/OS SNA initialization messages

```
ON BA01

"BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="KO1M2EQ"
"BSS1_GetEnv") KDCFP_TPNAME=KDCFC_TPNAME=KLXBS_TPNAME="SNASOCKETS"
"BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
,"getEnv") AF_SNA configuration: Alias(KO1M2EQ) Mode(CANCTDCS) TpName(SNASOCKETS)
........
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe

ON BA02

"BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="KO2M2EQ"
"BSS1_GetEnv") KDCFP_TPNAME=KDCFC_TPNAME=KLXBS_TPNAME="SNASOCKETS"
"BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
,"getEnv") AF_SNA configuration: Alias(KO2M2EQ) Mode(CANCTDCS) TpName(SNASOCKET)
.......
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe
```

Example 2-7 is based on Tivoli OMEGAMON XE for Mainframe Network (Tivoli Enterprise Monitoring Server) RKVLOG messages.

Example 2-7 z/OS Tivoli OMEGAMON XE for Mainframe Network SNA initialization messages

```
ON BA01

"BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="KO1N3NC"
"BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
"getEnv") AF_SNA configuration: Alias(KO1N3NC) Mode(CANCTDCS) TpName(SNASOCKETS)
.........
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe

ON BA02

"BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="KO2N3NC"
"BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
,"getEnv") AF_SNA configuration: Alias(KO2N3NC) Mode(CANCTDCS) TpName(SNASOCKETS)
........
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe
```

2.2 The distributed environment

This section discusses the design and implementation of the distributed part of the IBM Tivoli Monitoring environment.

2.2.1 Overview of the components

This section provides an overview of the main IBM Tivoli Monitoring components implemented in our environment.

Component description

The distributed environment comprises the following components:

- Tivoli Enterprise Monitoring Server can be considered as the central repository for all monitoring data. It performs the following functions:
  - Stores the definitions for conditions that indicate a problem with a particular resource
  - Controls the security of the monitoring solution
  - Each enterprise monitoring solution must contain one Tivoli Enterprise Monitoring Server Hub, and can include multiple Tivoli Enterprise Monitoring Server Remotes, which are used to scale large installations.
  - Tivoli Enterprise Monitoring Server Remote collects data from the agent and forwards it to the hub monitoring server.
  - Tivoli Enterprise Monitoring Server Hub correlates the monitoring data collected by the agents and the remote servers, and passes it to the Tivoli Enterprise Portal Server for presentation and evaluation.
- Tivoli Enterprise Portal Server functions as a repository for all the user data, such as the user IDs and user access control for the monitoring data, meaning the data each user can access and how it is displayed.
  - It connects to the Tivoli Enterprise Monitoring Server Hub and can be accessed by the Tivoli Enterprise Portal clients.
  - It provides a consistent look and feel for the users.
  - The database to store Tivoli Enterprise Portal server information is the DB2 Universal Database (UDB). The Structured Query Language (SQL) server is also supported.
- Tivoli Enterprise Portal provides a single point of control for managing the resources that the applications rely on, including a range of operating systems, servers, databases, platforms, and Web components.
- Tivoli Enterprise Monitoring Agent
  - The agents are the data collectors.
  - The agents monitor systems, subsystems, or applications, collect data, and pass the data to the Tivoli Enterprise Portal Server through the Tivoli Enterprise Monitoring Server.
  - The agents pass commands from the user to the system, subsystem, or application.
  - An agent interacts with a single system or application, and in most cases, resides on the same machine where the system or application is running.

Component model

The component model shows the static structure of the main elements that make up our IBM Tivoli Monitoring environment. This is shown in Figure 2-30.

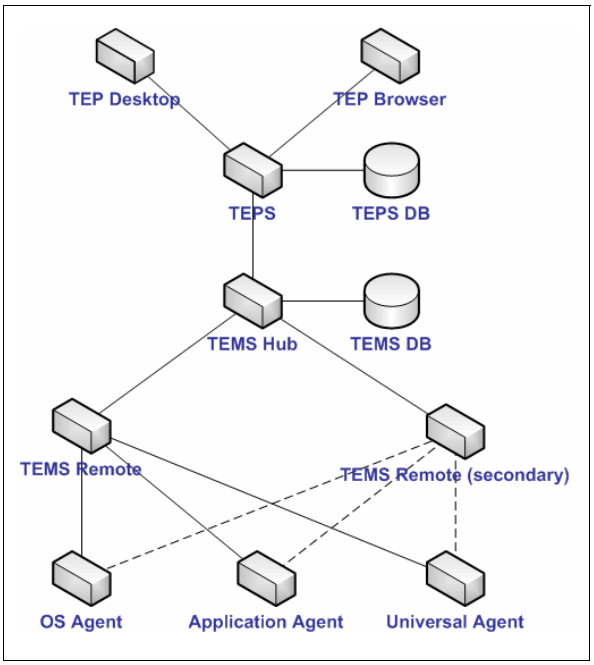


Figure 2-30 IBM Tivoli Monitoring component model

2.2.2 Platform description

The core of the IBM Tivoli Monitoring environment is based on the operating systems, as described in Table 2-1.

Table 2-1 IBM Tivoli Monitoring servers: Platform description

IBM Tivoli Monitoring component	Host name	Operating system	Comment
Tivoli Enterprise Monitoring Server Hub	vipa	z/OS	Refer to 2.1, “The IBM System z environment” on page 53 to get more information about the z/OS environment
Tivoli Enterprise Monitoring Server Remote	blade11	Windows 2003 SE Service Pack 1 (SP1)	► Primary Tivoli Enterprise Monitoring Server for Windows platforms ► Secondary Tivoli Enterprise Monitoring Server for UNIX platforms
Tivoli Enterprise Monitoring Server Remote	blade7	IBM AIX 5.2	► Primary Tivoli Enterprise Monitoring Server for UNIX platforms ► Secondary Tivoli Enterprise Monitoring Server for Windows platforms
Tivoli Enterprise Portal Server	blade8	Windows 2003 SE SP1	

The IBM Tivoli Monitoring environment must manage the following platforms:

- SUSE Linux Enterprise Server 9 Intel
- SUSE Linux Enterprise Server 9 for IBM eServer zSeries® 64-bit
- SUSE Linux Enterprise Server 9 for IBM eServer zSeries 31-bit
- AIX 5.2 64-bit
- Windows 2003 SE SP1 32-bit

2.2.3 Overview of network topology

The network comprises two main segments. The first segment is dedicated to the intranet communications between the client computers and the servers. The other segment is dedicated to the interserver communications, and is called *Production Network*. Most of the servers have two network interfaces connected on each segment or have only one connection to the Production Network.

Figure 2-31 shows the connection of the main IBM Tivoli Monitoring components to the production and the intranet networks. In particular, note the following points:

- All the IBM Tivoli Monitoring servers are physically connected to both the segments
- All the IBM Tivoli Monitoring servers are connected with gigabit interfaces
- There is no routing or bridging between the production network and the intranet network. These networks are isolated from one another.
- There is no firewall between the IBM Tivoli Monitoring servers and agents.
- No network address translation is implemented

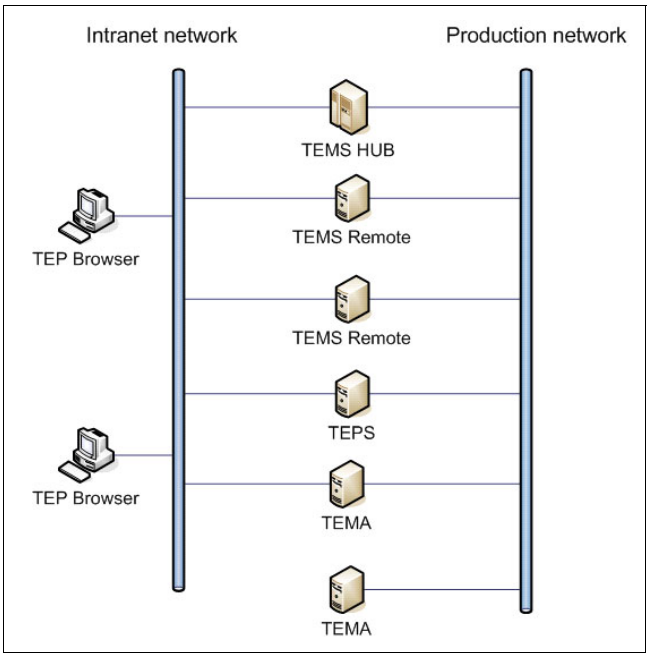


Figure 2-31 Network topology overview

2.2.4 Network communication

Network communication depends on the protocol used in the environment and the facilities offered by the hardware.

Communication protocol settings

The protocols selected for communication between the IBM Tivoli Monitoring components are based on the following requirements:

- Encryption is not required, considering that all the communication between the IBM Tivoli Monitoring components are on a protected production network that is unreachable directly from the intranet or other external sources.
- The Tivoli Enterprise Monitoring Server Remotes must communicate with the DVIPA of the z/OS system about where the Tivoli Enterprise Monitoring Server Hub is located. For such communication, we recommend IP.PIPE.
- A Moveable Hub solution is implemented on the z/OS. Therefore, the distributed environment must use port 2018 instead of the default port (1918) after verifying that 2018 is not already used for other applications.
- Due to the network topology, several servers are equipped with multiple network interface cards (NIC). There is a necessity to easily trace the network connections between the miscellaneous components.

Following are the selected protocols for Tivoli Enterprise Monitoring Server and Tivoli Enterprise Monitoring Agent communications:

- IP.PIPE

  This is the name given to the TCP protocol used for the Remote Procedure Call (RPC) packet transmission. This connection-oriented transport layer provides reliable and sequenced stream data delivery.

  Note: On a single system, IP.PIPE can share a maximum of 16 processes per listening port. This limitation occurs when running large amounts of Tivoli Enterprise Monitoring Agents on one physical system, such as running more than 16 Universal Agents or more than 16 Database Agent instances. Any processes above 16 will fall back to using the IP.UDP protocol. It is, however, not a limitation if the total amount of Tivoli Enterprise Monitoring Agents connect to one Tivoli Enterprise Monitoring Server.

  For details regarding the IP.PIPE protocol, refer to *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143.

- UDP.PIPE

  This is a connectionless, datagram protocol. User Datagram Protocol (UDP) performs faster and uses fewer resources. In large environments, if there is no security constraint, this protocol can be implemented as the first protocol for communication between Tivoli Enterprise Monitoring Agents and Tivoli Enterprise Monitoring Servers Remote.

  However, IP.PIPE must be used for communication between Tivoli Enterprise Monitoring Servers Remote and the Tivoli Enterprise Monitoring Server Hub when it runs on z/OS and uses the VIPA.

Table 2-2 shows the listening port setup of each component in our environment.

Table 2-2 IBM Tivoli Monitoring communication: Port usage

IBM Tivoli Monitoring component	Protocol	Listening port	Comment
Tivoli Enterprise Portal Server	► Hypertext Transfer Protocol (HTTP) ► HTTP over SSL	► 1920/tcp ► 15001/tcp	Listening ports to communicate with Tivoli Enterprise Portal
Tivoli Enterprise Monitoring Server Hub (z/OS)	► IP.PIPE ► IP.UDP	► tcp/2018 ► udp/2018	Listening ports to communicate with Tivoli Enterprise Monitoring Server Remote and Tivoli Enterprise Portal Server. The ports do not use the default value (1918) due to the implementation of the Moveable Hub on the z/OS platform.
Tivoli Enterprise Monitoring Server Remote	► IP.PIPE ► IP.UDP	► tcp/2018 ► udp/2018	Listening ports to communicate with Tivoli Enterprise Monitoring Agent
Tivoli Enterprise Monitoring Agent	► IP.PIPE ► IP.UDP	Calculated during Tivoli Enterprise Monitoring Agent startup	Listening port = 2018 + N x 4096, where N is the startup sequence

Communication model: Network connections

Figure 2-32 shows the connectivity using IP.PIPE and IP.UDP. The arrows indicate the direction of the connection established during the initialization phase of each component.

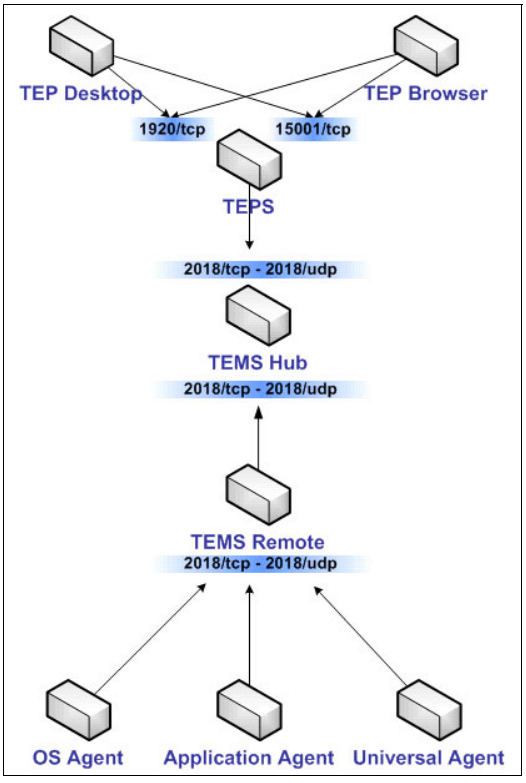


Figure 2-32 Communication model: Network connections (TCP environment)

Multiple network interface cards

In our environment, all the Tivoli Monitoring Servers and some Tivoli Monitoring Agents have more than one NIC. Therefore, for effective communication between the Tivoli Monitoring components, it is mandatory to identify which NIC to use. KDCB0_HOSTNAME enables you to specify which host name or TCP/IP address is bound to the NIC.

When specifying the KDCB0_HOSTNAME variable, the IBM Tivoli Monitoring components listen on all the interfaces. However, when initiating a connection to a remote component, the components use only the NIC specified in the KDCB0_HOSTNAME variable.

To enable the KDCB0_HOSTNAME variable, perform the following tasks:

1. Add the KDCB0_HOSTNAME variable in the configuration files used by the miscellaneous Tivoli Monitoring products. Table 2-3 shows the names of the config files in which the KDCB0_HOSTNAME variable is added.

Table 2-3 Config files to add KDCB0_HOSTNAME variable

Component	Variable	Config file
Windows Tivoli Enterprise Monitoring Server	KDCB0_HOSTNAME=<production_network_id>.<server_id>	<ITM_Install_dir>\cms\KBBENV
UNIX Tivoli Enterprise Monitoring Server	KDCB0_HOSTNAME=<production_network_id>.<server_id>	<ITM_Install_dir>/config/kbbenv.ini
Windows Tivoli Enterprise Portal Server	KDCB0_HOSTNAME=<production_network_id>.<server_id>	<ITM_Install_dir>\cnps\KFWENV
UNIX or Linux agents	KDCB0_HOSTNAME=<production_network_id>.<server_id>	<ITM_Install_dir>/config/env.config (read by all products in $CANDLEHOME. Therefore, all the agents installed on the servers use the value defined in KDCB0_HOSTNAME). Otherwise update the file <ITM_Install_dir>/config<product_code>.ini
Windows operating system (OS) agent	KDCB0_HOSTNAME=<production_network_id>.<server_id>	<ITM_Install_dir>\tmaitm6\KNTENV
Other agents on Windows	KDCB0_HOSTNAME=<production_network_id>.<server_id>	<ITM_Install_dir>\[tbd]\K<product_code>ENV.ini

2. When the config file is updated, recycle the Tivoli Enterprise Monitoring Server or the Agent.
3. When a Tivoli Enterprise Monitoring Server or Agent starts, if the KDCB0_HOSTNAME variable is enabled, a message is retrieved in the log file as shown in Example 2-8.

Example 2-8 KDCB0_HOSTNAME activation in the Tivoli Enterprise Monitoring Server log

```
Monday, February 27, 2006, 19:12:29-{AE4}kbbssge.c,52,"BSS1_GetEnv")
KDCB0_HOSTNAME="10.1.1.135"
(Monday, February 27, 2006, 19:12:29-{AE4}kdebprc.c,676,"interface_discovery")
IPV4 interface list: '10.1.1.135'
```

2.2.5 Estimating the capacity of the monitoring environment

Estimating the capacity of the monitoring environment helps determine a margin or boundary before considering that the environment is no longer adapted to manage the workload correctly. Of course, this is not a strict margin because too many factors, such as network bandwidth, hardware limitation, and so on, may influence the environment's behavior.

Following is the maximum capacity of a monitoring environment based on one Tivoli Enterprise Monitoring Server Hub:

- The number of Tivoli Enterprise Monitoring Servers Remote supported: 15
- The number of Agents per Tivoli Enterprise Monitoring Server Remote: 500
- The number of heartbeating agents per Tivoli Enterprise Monitoring Server: 500
- The number of consoles connected to the Tivoli Enterprise Portal Server: 50
- The number of active consoles (mining data) connected to the Tivoli Enterprise Portal Server: 10

  Note: This requirement can also be stated as 50 active consoles on which users access real-time data 20% of the time.

- Maximum number of total agents: 5,000

  Note: The number of agents may vary on each server. Most agents perform a specific monitoring function (OS, DB, applications). The Universal Agent can manage several types of monitoring functions (files, snmp, and so on).

Based on these recommendations, our environment has the following capacity limit:

- Maximum number of agents for the distributed environment (two Tivoli Enterprise Monitoring Servers Remote): 2 x 500
- Maximum number of agents for the z/OS environment (two Tivoli Enterprise Monitoring Servers Remote): 2 x 500

Two Tivoli Enterprise Monitoring Servers Remote are implemented in the distributed environment to enable a failover solution in case one Tivoli Enterprise Monitoring Server Remote fails. Therefore, the maximum number of agents for the distributed environment must be limited to 500. However, for a larger environment, we recommend that you dedicate only one Tivoli Enterprise Monitoring Server Remote for failover. (This Tivoli Enterprise Monitoring Server does *not* manage any agent during normal condition.)

For more information about IBM Tivoli Monitoring environment sizing, refer to *Deployment Guide Series: IBM Tivoli Monitoring 6.1*, SG24-7188.

2.2.6 Failover support

The global availability depends on the availability of the Hub and the availability of the agents.

- IBM Tivoli Monitoring provides a hot standby feature to enable the continuous availability of the Tivoli Enterprise Monitoring Server Hub. However, in our environment, the failover of the Tivoli Enterprise Monitoring Server Hub is managed in a different way based on the z/OS environment capabilities. Refer to 2.1, “The IBM System z environment” on page 53 for details.
- Agents can be configured to connect to a secondary Tivoli Enterprise Monitoring Server to ensure availability. If the Tivoli Enterprise Monitoring Server to which agents are connected becomes unavailable, the agents switch to the secondary Tivoli Enterprise Monitoring Server. The agents exhibit the following behavior:
  - When the communication with the primary Tivoli Enterprise Monitoring Server cannot be established, agents automatically switch to the secondary Tivoli Enterprise Monitoring Server.
  - When the primary Tivoli Enterprise Monitoring Server becomes available, there is no automatic fallback of agents to the primary Tivoli Enterprise Monitoring Server.

In our environment, both the Tivoli Enterprise Monitoring Servers Remote are known by the agents and are able to handle the communication coming from the agents for which the primary Tivoli Enterprise Monitoring Server Remote is unavailable.

Based on the heartbeat mechanism, agents try every 10 minutes, which is the default value, to re-establish Tivoli Enterprise Monitoring Server communication when the connection fails. When a secondary Tivoli Enterprise Monitoring Server is defined, agents connect to the secondary Tivoli Enterprise Monitoring Server when the communication for the heartbeat update has failed with the primary Tivoli Enterprise Monitoring Server. Refer to 2.2.6, “Failover support” on page 91 for details about the heartbeat behavior and the setup.

The heartbeat mechanism is used to monitor the status of the remote monitoring servers and the monitoring agents. Each Tivoli Enterprise Monitoring Server expects a heartbeat timestamp from the agents every 10 minutes (default value). If no heartbeat is received in this time range, it concludes that the agent is offline. Figure 2-33 depicts the heartbeat behavior.

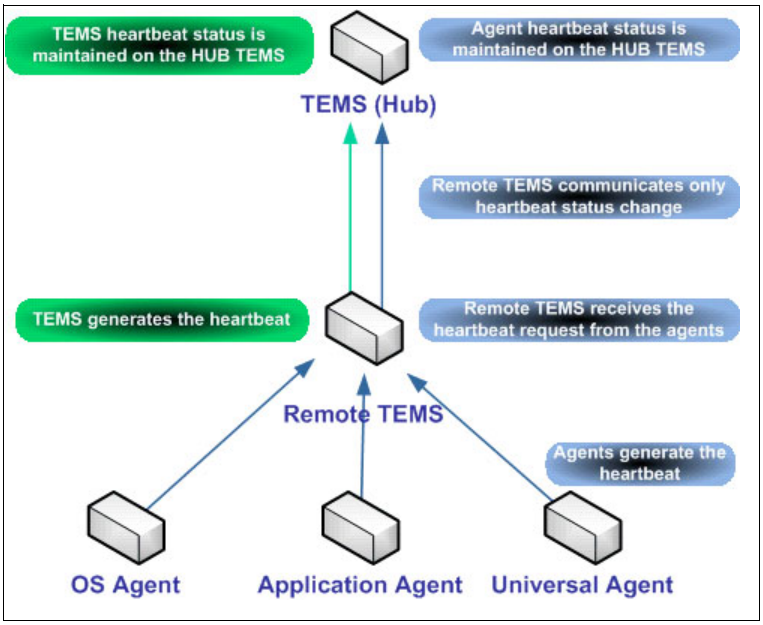

TEMS heartbeat status is maintained on the HUB TEMS
Agent heartbeat status is maintained on the HUB TEMS
TEMS (Hub)
Remote TEMS communicates only heartbeat status change
TEMS generates the heartbeat
Remote TEMS receives the heartbeat request from the agents
Remote TEMS
Agents generate the heartbeat
OS Agent
Application Agent
Universal Agent

Figure 2-33 IBM Tivoli Monitoring heartbeat behavior

The main features of the heartbeat are:

- Default heartbeat frequency for agents is 10 minutes
- Default heartbeat frequency for Tivoli Enterprise Monitoring Server is 3 minutes
- Heartbeat frequency must be set in the agent and in the Tivoli Enterprise Monitoring Server. If no heartbeat interval is set, the default value is used.
- Agents communicate the heartbeat interval during the initial heartbeat request sent to the Tivoli Enterprise Monitoring Server
- Each agent generates its own heartbeat

Table 2-4 shows the heartbeat parameters defined in our distributed environment to detect when an agent becomes unavailable. In order to avoid heartbeat overflow, only OS agents have a heartbeat value that is lower than the default value.

Table 2-4 Heartbeat parameters

Component	Value (minutes)	Config file
Windows Tivoli Enterprise Monitoring Server	CTIRA_HEARTBEAT=3	KBBENV
UNIX Tivoli Enterprise Monitoring Server	CTIRA_HEARTBEAT=3	kbbenv.ini
UNIX OS agent[a]	CTIRA_HEARTBEAT=5	ux.ini
Linux OS agent[b]	CTIRA_HEARTBEAT=5	lz.ini
Windows OS agent	CTIRA_HEARTBEAT=5	KNTENV
Other agents on UNIX	CTIRA_HEARTBEAT=10	<product_code>.ini
Other agents on Windows	CTIRA_HEARTBEAT=10	K<product_code>ENV.ini

a. Add a parameter to the ux.ini file before the agents are configured. Otherwise, they are *not* invoked when the agents are started.
b. Add a parameter to the lz.ini file before the agents are configured. Otherwise, they are *not* invoked when the agents are started.

Note: Add or modify the KDC_HEARTBEAT parameter to the agents' ini files. Otherwise, the configuration file are *not* updated.

2.2.7 Operating system-specific considerations

After installing the OS on the servers dedicated to run the IBM Tivoli Monitoring components such as Tivoli Enterprise Portal Server and Tivoli Enterprise Monitoring Server, ensure that the systems have the suitable parameters to run IBM Tivoli Monitoring efficiently. Even with powerful hardware, the system may face performance issues due to a bad setup of the OS parameters.

Tuning the Windows operating system

After completing the OS installation, perform the following tasks:

1. Right-click **My Computer** and select **Properties** → **Advanced** → **Performance**. Check the **Processor Scheduling** and **Memory Usage** parameters.
2. The window shown in Figure 2-34 opens. The Virtual Memory field in the window is the size of the paging file and reflects the amount of disk space that must be reserved beyond the real storage that is available on the machine. Windows sets this parameter based on real storage size. Change the value by clicking **Change**.

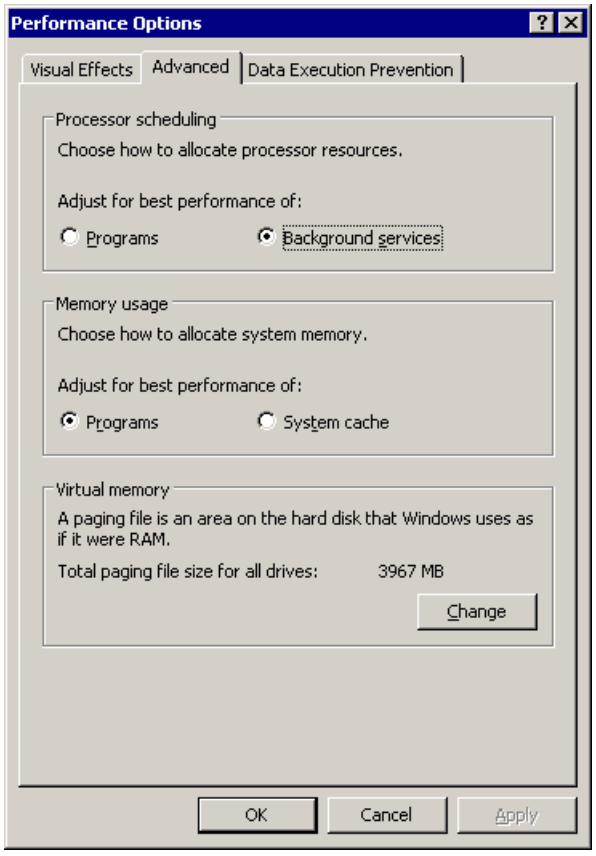


Figure 2-34 Windows performance options

3. Return to your desktop and select **Start** → **My Network Places**.
   a. Right-click **My Network Places** and select **Properties**.
   b. Right-click one of the available network cards and select **Properties**.
   c. Select **File and Printer Sharing for Microsoft Networks**.
   d. Click **Properties**.
   e. Check the relevant Server Optimization parameter, as shown in Figure 2-35.

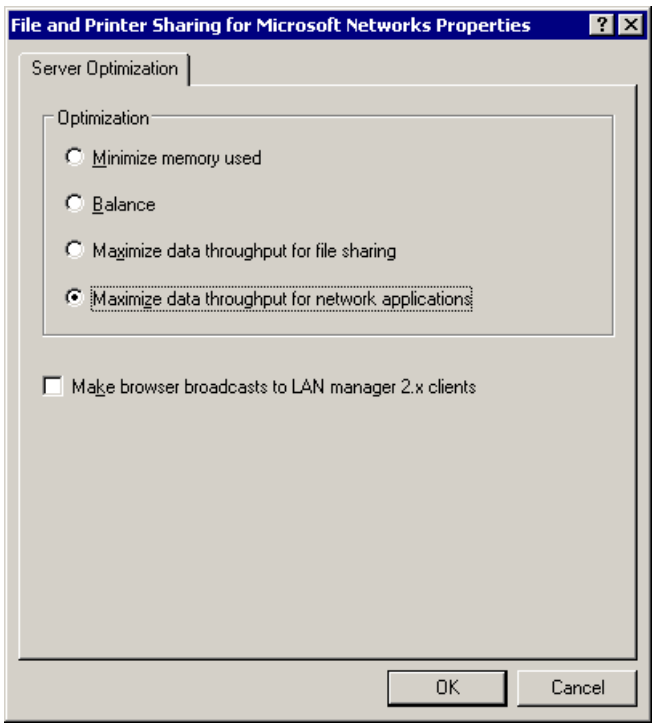


Figure 2-35 Windows network server optimization

Tuning the AIX operating system

Verify the size of the data area by running the **`ulimit -d`** command, where the -d option specifies the size of the data area in kilobytes. If the setting returned is less than 256 MB, increase the maxfiles limit to at least 256 MB.

2.2.8 Database considerations

The database implemented for the Tivoli Enterprise Portal Server is IBM DB2 Enterprise Edition 8.1.11 for Windows. This database is installed locally on the server where the Tivoli Enterprise Portal Server is installed. In our case, the default installation process proposed by the DB2 Installer is used without any specific customization. A basic tuning of the database is carried out to contribute to the overall performance of the Tivoli Enterprise Portal Server. This tuning is based on the recommendations available in the DB2 Administration Guide. In particular, we tuned the buffer pool. The configuration of the buffer pool is the single-most important tuning area because the delay caused by slow input/output (I/O) can be reduced.

A buffer pool is the amount of main memory allocated to the cache table and the index data pages when they are read from the disk or are being modified. The purpose of the buffer pool is to improve system performance. Data can be accessed much faster from memory than from disk. Therefore, the fewer the number of times the database manager has to read from or write to a disk (I/O), the better the performance. Each table space is associated with a

specific buffer pool. The default buffer pool, IBMDEFAULTBP, is used by Tivoli Enterprise Portal Server. By increasing the default value, the number of logical access is high when compared to the number of physical access (direct disk I/O). Therefore, there is a low frequency of disk I/O.

Table 2-5 provides the buffer pool size defined for the Tivoli Enterprise Portal Server database.

Table 2-5 DB2: Buffer pool value for the Tivoli Enterprise Portal Server

Table space	Buffer pool	Number of pages	Page size (KB)	Memory usage (MB)
USERSPACE1	IBMDEFAULTBP	15000	4	60

Note: The default value for the IBMDEFAULTBP buffer pool is as follows:

- 250 on Windows
- 1000 on UNIX

In order to increase the buffer pool size from the DB2 command prompt, perform the following tasks:

- Connect DB2 to the Tivoli Enterprise Portal Server
- Change the IBMDEFAULTBP parameter to 15000

2.2.9 Installing the monitoring servers

In our case, the basic installation process of the Tivoli Enterprise Monitoring Server remote and Tivoli Enterprise Portal Server was performed based on the recommendations available in *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407 and in *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143. We recommend that you also refer to Chapter 2.2.4, “Network communication” on page 87 regarding the communication ports used for communication between the IBM Tivoli Monitoring components. This chapter explains the specific post-installation stages to customize the IBM Tivoli Monitoring servers.

Customizing the Tivoli Enterprise Monitoring Server Remote

This section describes the miscellaneous tasks involved in customizing the IBM Tivoli Monitoring Servers Remote to address the requirements of our environment. Following is a list of these tasks:

- Defining the network interface to communicate with Tivoli Enterprise Monitoring Server Hub and agents

  After completing the basic installation, because all the Tivoli Enterprise Monitoring Servers have several NICs, it is necessary to specify the one used by Tivoli Enterprise Monitoring Server. Refer to “Multiple network interface cards” on page 89 for details.

- Application support for Windows and Linux

  On the AIX Tivoli Enterprise Monitoring Server Remote, it is necessary to add agent-specific information to the monitoring server. This task was earlier called seeding operation. Without this operation, all the OS agents connected to this Tivoli Enterprise Monitoring Server, except AIX platforms, cannot be queried from the Tivoli Enterprise Portal Server. The workspaces are empty and an error message is returned.

To add the agent-specific information for Windows and Linux, execute the following from the UNIX command prompt:

```
itmcmd support -t <TEMS_name> nt lz
```

- Adding application support for distributed applications on Tivoli Enterprise Monitoring Servers

  The Tivoli OMEGAMON XE agents are not built specifically for IBM Tivoli Monitoring. However, they can interact with the IBM Tivoli Monitoring environment by enabling the application support (ATTR and CAT files).

  Download the application support for distributed applications from the following Web site after providing a user ID and password:

  https://www14.software.ibm.com/webapp/iwm/web/reg/download.do?source=tmdl&S_PKG=dl&cp=UTF-8

  From this Web site, download the application installer available for miscellaneous platforms (Windows, Linux, and so on) and a package containing the application support for the OMEGAMON products.

  To install the application support on Windows Tivoli Enterprise Monitoring Server, perform the following tasks:

  a. Run the application installer using the **setupwin32.exe** command.

  b. In the Welcome page that is displayed, click **Next**.

  c. Enter the IBM Tivoli Monitoring installation directory, specify the location of application support files downloaded from the Web site, as shown Figure 2-36, and click **Next**.

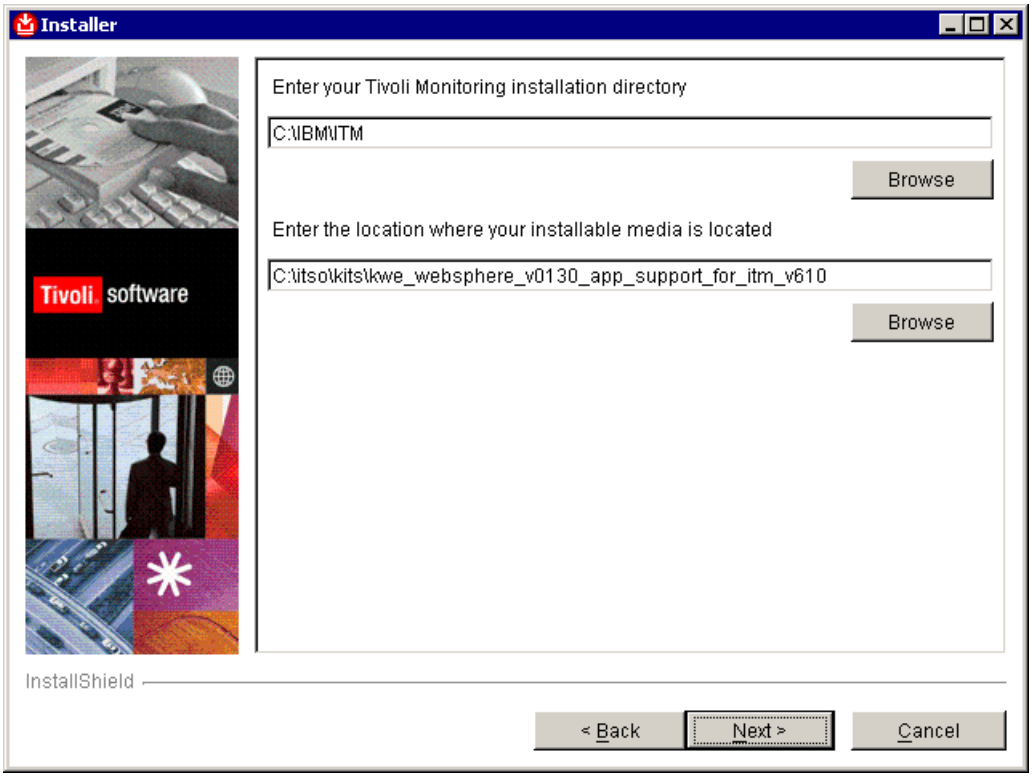


Figure 2-36 Application support for WebSphere Application Server (distributed): Directories location

  d. Usually, the installation program detects that the installation is executed on a Tivoli Enterprise Monitoring Server and the check box against Tivoli Enterprise Monitoring Server is selected. Click **Next**.

e. Select the application for which you want to install the application support files. Click **Next**.

f. The Installer (Figure 2-37) displays the installation actions that are performed. Click **Next**.

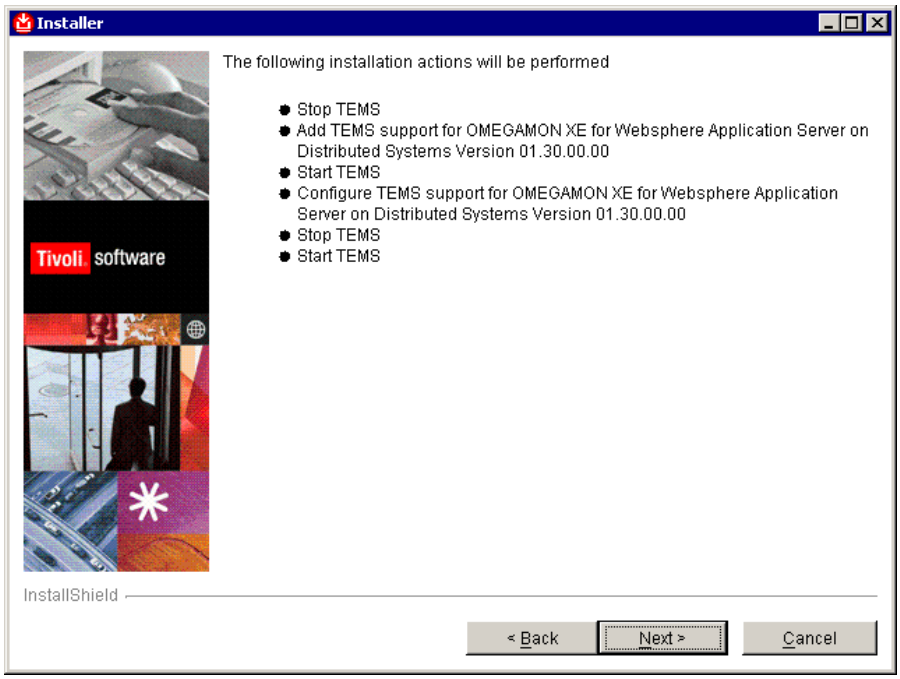

Installer
The following installation actions will be performed
Stop TEMS
Add TEMS support for OMEGAMON XE for Websphere Application Server on Distributed Systems Version 01.30.00.00
Start TEMS
Configure TEMS support for OMEGAMON XE for Websphere Application Server on Distributed Systems Version 01.30.00.00
Stop TEMS
Start TEMS
Tivoli software
InstallShield
< Back
Next >
Cancel

Figure 2-37 Application support for WebSphere Application Server (distributed): Installation stages

g. After the application support files are installed, click **Finish**.

h. Figure 2-38 shows the installation completion window.

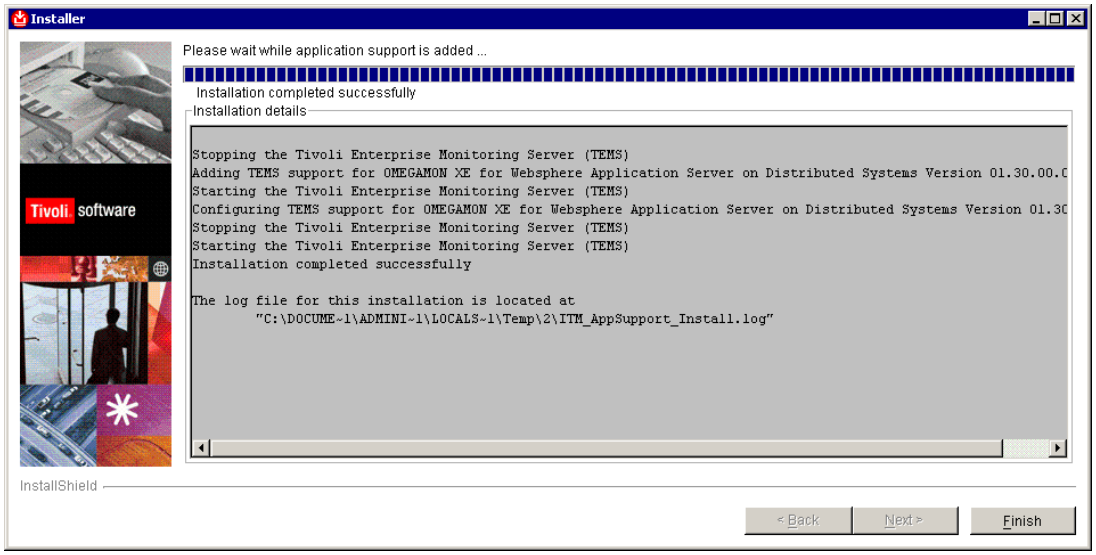

Installer
Please wait while application support is added ...
Installation completed successfully
Installation details
Stopping the Tivoli Enterprise Monitoring Server (TEMS)
Adding TEMS support for OMEGAMON XE for Websphere Application Server on Distributed Systems Version 01.30.00.0
Starting the Tivoli Enterprise Monitoring Server (TEMS)
Configuring TEMS support for OMEGAMON XE for Websphere Application Server on Distributed Systems Version 01.30
Stopping the Tivoli Enterprise Monitoring Server (TEMS)
Starting the Tivoli Enterprise Monitoring Server (TEMS)
Installation completed successfully
The log file for this installation is located at
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\2\ITM_AppSupport_Install.log"
Tivoli software
InstallShield
< Back
Next >
Finish

Figure 2-38 Application support for WebSphere Application Server (distributed): Installation completion

Customizing the Tivoli Enterprise Portal Server

This section describes the miscellaneous tasks involved in customizing the Tivoli Enterprise Portal Server in order to address the requirements of our environment. (This section does not discuss the Tivoli Enterprise Monitoring Server requirements. They are described in the installation documentation.) Following is a list of the miscellaneous tasks:

- Basic tuning of the database

  Some of the default parameters defined during the DB2 installation must be modified to perform a basic tuning.

- Defining the network interface to communicate with Tivoli Enterprise Monitoring Server Hub

  After completing the basic installation, because the Tivoli Enterprise Portal Server has several NICs, it is necessary to specify the one used by IBM Tivoli Monitoring. Refer to “Multiple network interface cards” on page 89 for details.

- Defining the network interface to communicate with Tivoli Enterprise Portal

  The Tivoli Enterprise Portal clients (desktop and browser) are located on the intranet network, as specified in “Overview of network topology” on page 86. Because IBM Tivoli Monitoring servers are configured to communicate only on the production network and there is no routing or bridging between the two networks, the Tivoli Enterprise Portal Server is unable to correctly handle the requests coming from the Tivoli Enterprise Portal clients (desktop and browser). Any Tivoli Enterprise Portal client trying to open a session is able to get the login dialog box, but after entering the password, the user receives the following error message:

  ```
  KFWITM215E Unable to process logon request.
  ```

Perform the following tasks to enable the Tivoli Enterprise Portal Server to handle the connections coming from the intranet network:

a. Launch the Manage Tivoli Monitoring Services program and right-click the relevant option, in this case, the Tivoli Enterprise Portal Server, as shown in Figure 2-39. Select **Advanced** → **Configure TEPS Interfaces**.

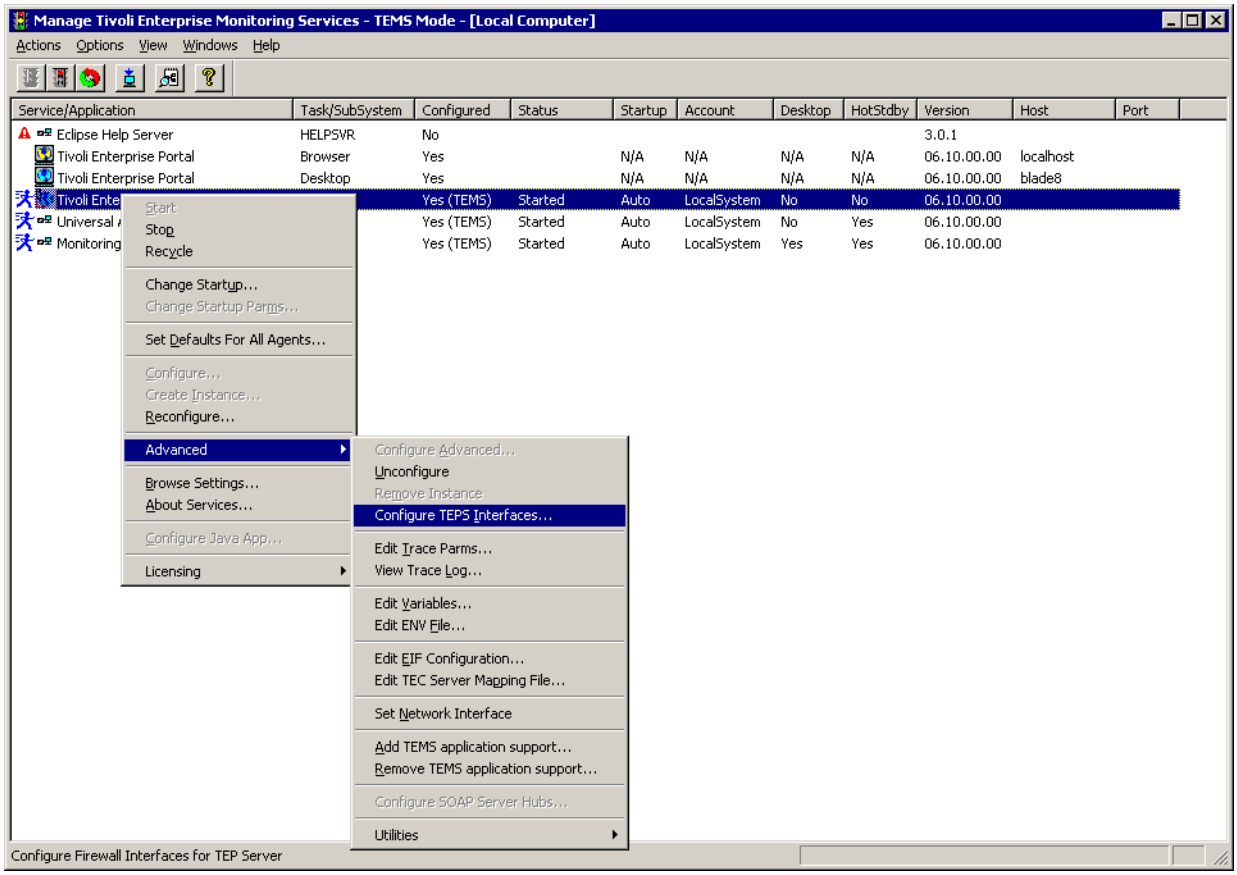

Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]
Actions Options View Windows Help
Service/Application Task/SubSystem Configured Status Startup Account Desktop HotStdby Version Host Port
Eclipse Help Server HELPSVR No 3.0.1
Tivoli Enterprise Portal Browser Yes N/A N/A N/A N/A 06.10.00.00 localhost
Tivoli Enterprise Portal Desktop Yes N/A N/A N/A N/A 06.10.00.00 blade8
Tivoli Ente Yes (TEMS) Started Auto LocalSystem No No 06.10.00.00
Universal A Yes (TEMS) Started Auto LocalSystem No Yes 06.10.00.00
Monitoring Yes (TEMS) Started Auto LocalSystem Yes Yes 06.10.00.00
Start
Stop
Recycle
Change Startup...
Change Startup Parms...
Set Defaults For All Agents...
Configure...
Create Instance...
Reconfigure...
Advanced
Browse Settings...
About Services...
Configure Java App...
Licensing
Configure Advanced...
Unconfigure
Remove Instance
Configure TEPS Interfaces...
Edit Trace Parms...
View Trace Log...
Edit Variables...
Edit ENV File...
Edit EIF Configuration...
Edit TEC Server Mapping File...
Set Network Interface
Add TEMS application support...
Remove TEMS application support...
Configure SOAP Server Hubs...
Utilities
Configure Firewall Interfaces for TEP Server

Figure 2-39 Configuring the Tivoli Enterprise Portal Server interface menu

b. In the window that opens (Figure 2-40), update the Hostname or IP Address parameter. Because this one is usually left blank, the server uses the IP address that it detects as the server's primary address. If specified, it defines the interface that is to be used to communicate with the Tivoli Enterprise Portal client, as shown in Figure 2-40. Click **OK**, and refresh the Tivoli Enterprise Portal Server.

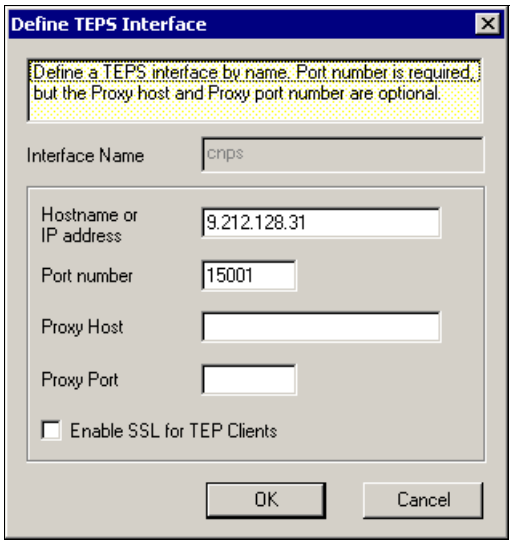


Figure 2-40 Tivoli Enterprise Portal Server interface parameter

- Avoiding password encryption

  When the integrated cryptographic service facility (ICSF) on z/OS is not activated, the Tivoli Enterprise Monitoring Server Hub is unable to decrypt the password that is sent by the Tivoli Enterprise Portal Server to identify a user. Therefore, the Tivoli Enterprise Monitoring user cannot be identified, and the login is rejected.

  To disable the encryption, perform the following tasks:

  a. In the Tivoli Enterprise Portal Server, edit the kfwenv configuration file located in the CNPS directory and add the following control statement:

     ```
     USE_EGG1_FLAG=1
     ```

  b. Refresh the Tivoli Enterprise Portal Server

- Adding application support for z/OS applications

  Ensure that the version of the products selected here match the versions installed on z/OS. If the installation process is run multiple times, ensure that the same version is selected each time. Otherwise, the data is corrupted and the product generates unpredictable results.

  The default installation of the Tivoli Enterprise Portal Server does not include the application support (ODI files) for managing the agents running on z/OS. Therefore, it is necessary to install application support for z/OS OMEGAMON V350 agent and V360 agent on the Tivoli Enterprise Portal Server. Because in our environment all the agents running on z/OS are connected to the Tivoli Enterprise Monitoring Servers Remote located on z/OS systems, it is not necessary to perform this task on the Tivoli Enterprise Monitoring Servers Remote installed on the AIX and Windows platforms. Remember that the Tivoli Enterprise Monitoring Server Hub runs on z/OS. Therefore, there is no necessity to update the application support on this server as well.

The CD, *Monitoring Services on z/OS 350/360 Workspace Enablement* (V6.1), contains the specific setup and packages to install the application support. Download the contents of this CD from the OMEGAMON v350/v360 Application Enablement Web site:

https://www14.software.ibm.com/webapp/iwm/web/reg/download.do?source=tmdl&S_PKG=dl&cp=UTF-8

A detailed Installation Guide is available on the CD or in the packages available on this Web site.

To install the application support on Windows Tivoli Enterprise Portal Server, perform the following tasks:

a. Run setupwin32.exe from the CD *Monitoring Services on z/OS 350/360 Workspace Enablement*.

b. During the installation process, specify which Tivoli Monitoring component you want to add application support to.

c. Select the Tivoli Enterprise Portal Server.

d. Select the application you want to add support for, as shown Figure 2-41.

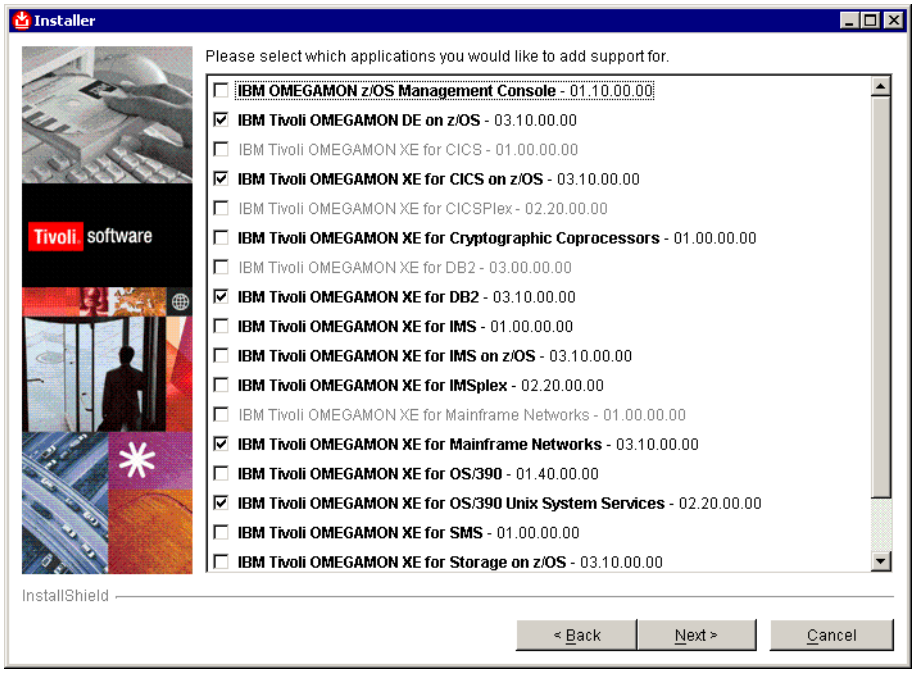

Installer
Please select which applications you would like to add support for.
IBM OMEGAMON z/OS Management Console - 01.10.00.00
IBM Tivoli OMEGAMON DE on z/OS - 03.10.00.00
IBM Tivoli OMEGAMON XE for CICS - 01.00.00.00
IBM Tivoli OMEGAMON XE for CICS on z/OS - 03.10.00.00
IBM Tivoli OMEGAMON XE for CICSPlex - 02.20.00.00
IBM Tivoli OMEGAMON XE for Cryptographic Coprocessors - 01.00.00.00
IBM Tivoli OMEGAMON XE for DB2 - 03.00.00.00
IBM Tivoli OMEGAMON XE for DB2 - 03.10.00.00
IBM Tivoli OMEGAMON XE for IMS - 01.00.00.00
IBM Tivoli OMEGAMON XE for IMS on z/OS - 03.10.00.00
IBM Tivoli OMEGAMON XE for IMSplex - 02.20.00.00
IBM Tivoli OMEGAMON XE for Mainframe Networks - 01.00.00.00
IBM Tivoli OMEGAMON XE for Mainframe Networks - 03.10.00.00
IBM Tivoli OMEGAMON XE for OS/390 - 01.40.00.00
IBM Tivoli OMEGAMON XE for OS/390 Unix System Services - 02.20.00.00
IBM Tivoli OMEGAMON XE for SMS - 01.00.00.00
IBM Tivoli OMEGAMON XE for Storage on z/OS - 03.10.00.00
Tivoli software
InstallShield
< Back
Next >
Cancel

Figure 2-41 z/OS application support selection

e. When the operation is completed, you must receive the messages displayed in the window, as shown in Figure 2-42.

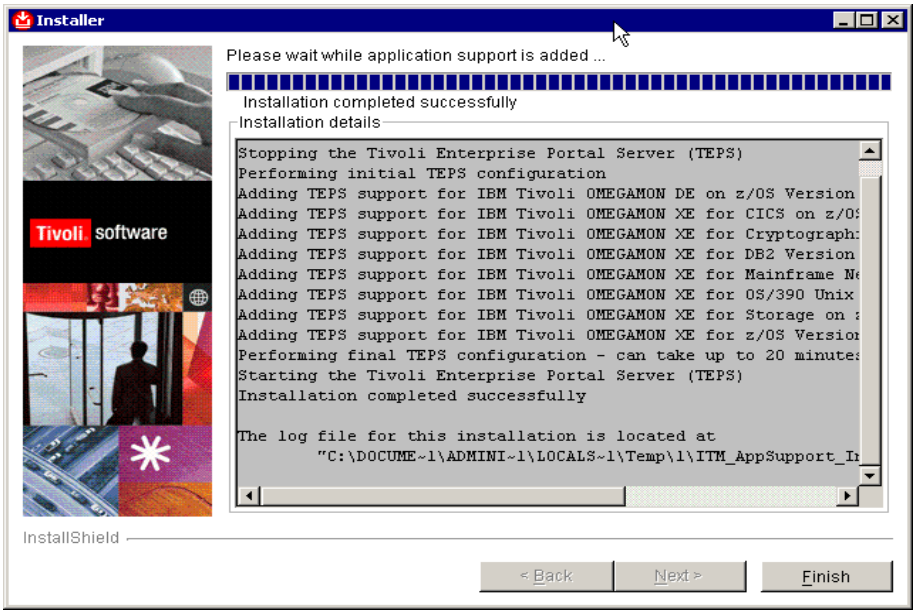

Installer
Please wait while application support is added ...
Installation completed successfully
Installation details
Stopping the Tivoli Enterprise Portal Server (TEPS)
Performing initial TEPS configuration
Adding TEPS support for IBM Tivoli OMEGAMON DE on z/OS Version
Adding TEPS support for IBM Tivoli OMEGAMON XE for CICS on z/O
Adding TEPS support for IBM Tivoli OMEGAMON XE for Cryptograph
Adding TEPS support for IBM Tivoli OMEGAMON XE for DB2 Version
Adding TEPS support for IBM Tivoli OMEGAMON XE for Mainframe N
Adding TEPS support for IBM Tivoli OMEGAMON XE for OS/390 Unix
Adding TEPS support for IBM Tivoli OMEGAMON XE for Storage on
Adding TEPS support for IBM Tivoli OMEGAMON XE for z/OS Versio
Performing final TEPS configuration - can take up to 20 minute
Starting the Tivoli Enterprise Portal Server (TEPS)
Installation completed successfully
The log file for this installation is located at
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1\ITM_AppSupport_I
Tivoli software
InstallShield
< Back
Next >
Finish

Figure 2-42 z/OS application support: Installation ends

- Adding application support for distributed applications on Tivoli Enterprise Portal Server

  As explained in “IBM Tivoli Enterprise Monitoring Server Hub” on page 107, Tivoli Enterprise Portal Server must be similar to the Tivoli Enterprise Monitoring Server Remote, updated with the application support (ODI files only) to interact with the Tivoli OMEGAMON XE agents used for the distributed applications.

  The installation process on the Tivoli Enterprise Portal Server is similar to the one used for the distributed application support described earlier in “IBM Tivoli Enterprise Monitoring Server Hub” on page 107.

  After the previous installation process is complete, it is essential to update the Tivoli Enterprise Monitoring Server Hub catalogs and attributes running on z/OS used for the Tivoli OMEGAMON XE agents. Without this update, the information related to these agents is not available in the Tivoli Enterprise Portal workspaces.

Perform this operation from the Tivoli Enterprise Portal Server:

a. Launch the Manage Tivoli Monitoring Services program and right-click **Tivoli Enterprise Portal Server** → **Utilities** → **FTP Catalog and Attribute Files**, as shown in Figure 2-43.

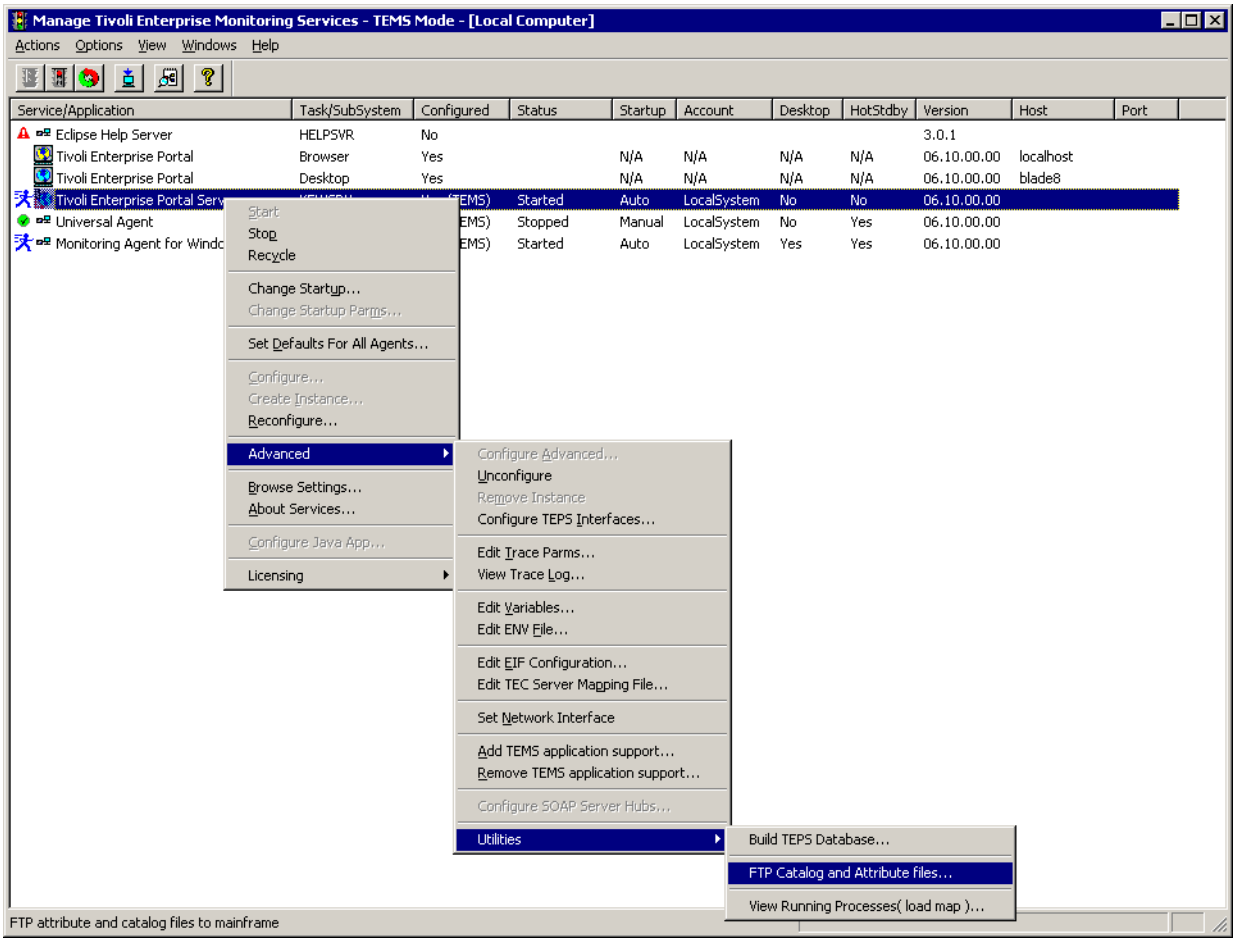

Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]
Actions Options View Windows Help
Service/Application Task/SubSystem Configured Status Startup Account Desktop HotStdby Version Host Port
Eclipse Help Server HELPSVR No 3.0.1
Tivoli Enterprise Portal Browser Yes N/A N/A N/A N/A 06.10.00.00 localhost
Tivoli Enterprise Portal Desktop Yes N/A N/A N/A N/A 06.10.00.00 blade8
Tivoli Enterprise Portal Serv EMS) Started Auto LocalSystem No No 06.10.00.00
Universal Agent EMS) Stopped Manual LocalSystem No Yes 06.10.00.00
Monitoring Agent for Windc EMS) Started Auto LocalSystem Yes Yes 06.10.00.00
Start
Stop
Recycle
Change Startup...
Change Startup Parms...
Set Defaults For All Agents...
Configure...
Create Instance...
Reconfigure...
Advanced
Browse Settings...
About Services...
Configure Java App...
Licensing
Configure Advanced...
Unconfigure
Remove Instance
Configure TEPS Interfaces...
Edit Trace Parms...
View Trace Log...
Edit Variables...
Edit ENV File...
Edit EIF Configuration...
Edit TEC Server Mapping File...
Set Network Interface
Add TEMS application support...
Remove TEMS application support...
Configure SOAP Server Hubs...
Utilities
Build TEPS Database...
FTP Catalog and Attribute files...
View Running Processes(load map)...
FTP attribute and catalog files to mainframe

Figure 2-43 File Transfer Protocol (FTP) catalog and attribute files selection

b. In the window that opens, select the attribute and catalog data for transfer (if you select all of them, you are presented with the option of either overwriting or not overwriting the files that already exist on z/OS). Click **OK**.

c. In the window that opens (Figure 2-44), specify the target, the privileged z/OS account, password, and the Data Set Name.

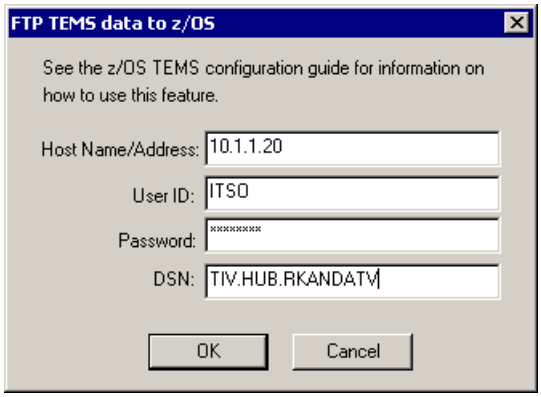


Figure 2-44 FTP of the Tivoli Enterprise Monitoring Server to the z/OS parameters

d. In the window that opens (Figure 2-45), click **No** to update only the new attribute and catalog files.

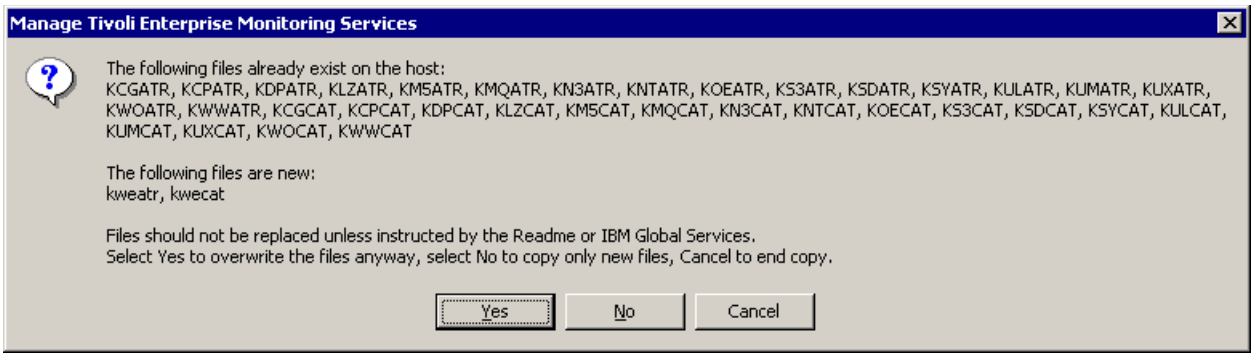

Manage Tivoli Enterprise Monitoring Services
The following files already exist on the host:
KCGATR, KCPATR, KDPATR, KLZATR, KM5ATR, KMQATR, KN3ATR, KNTATR, KOEATR, KS3ATR, KSDATR, KSYATR, KULATR, KUMATR, KUXATR, KWOATR, KWWATR, KCGCAT, KCPCAT, KDPCAT, KLZCAT, KM5CAT, KMQCAT, KN3CAT, KNTCAT, KOECAT, KS3CAT, KSDCAT, KSYCAT, KULCAT, KUMCAT, KUXCAT, KWOCAT, KWWCAT
The following files are new:
kweatr, kwecat
Files should not be replaced unless instructed by the Readme or IBM Global Services.
Select Yes to overwrite the files anyway, select No to copy only new files, Cancel to end copy.
Yes
No
Cancel

Figure 2-45 Confirmation of the FTP

e. When the transfer is complete, refresh the Tivoli Enterprise Monitoring Server Hub.

In our case, with the Hub running on z/OS and the environment provided by Tivoli Enterprise Monitoring Server, a manual seeding operation to get the proper information into the Hub is required. This involves selecting **Actions** → **Advanced** → **Add TEMS Application Support**. If this task is not performed, some of the products do not function.

2.2.10 Deploying the remote operating system agents

The remote deployment from the command prompt enables you to quickly distribute the OS agents and get them up and running from a central location. In our environment, all the OS agents are installed from the Windows Tivoli Enterprise Monitoring Server Remote. When the OS agent is available on a target server, it is possible to install other agents directly from the Tivoli Enterprise Portal or from the Tivoli Enterprise Monitoring Server command prompt.

Populating the deployment depot

Before beginning the installation, ensure that the packages are already in the depot server where the installation process is run. By default, our depot does not contain all the packages required to install the OS agents on the following platforms:

- SUSE Linux Enterprise Server 9 Intel
- SUSE Linux Enterprise Server 9 for IBM eServer zSeries 64-bit
- SUSE Linux Enterprise Server 9 for IBM eServer zSeries 31-bit
- AIX 5.2 64-bit

In order to add the OS agent for Linux on System z 64-bit, use the tacmd addBundles commands:

```
tacmd addbundles -i .<ITM_zlinux_CD>\unix -t lz -p ls3266
```

In this command, the parameters indicate the following:

- The parameter -i specifies the source of the installation images. Here, ITM_zlinux_CD is the root of the CD-ROM containing the IBM Tivoli Monitoring images for Linux on System z.
- The parameter -t specifies the agent product code (lz for Linux, ux for UNIX, and so on)
- The parameter -p specifies the host type. Generally, the host type value provides the following information:
  - Product name: aix for AIX, sol for Solaris, ls for SUSE Linux, and so on
  - Product version: 51 for AIX 5.1, 2.6 for Linux with 2.6 kernel, and so on
  - Platform type: 3 for 32 bits architecture, 6 for 62 bits architecture

Running the deployment

To quickly deploy the OS agents, a simple batch script running on Windows is created to address the following requirements:

- The agent installation is done by the tacmd createNode command
- The agent is able to connect on the primary or the secondary Tivoli Enterprise Monitoring Server Remote (backup).
- The agent uses the port 2018 (default is 1918) to connect to the Tivoli Enterprise Monitoring Server Remote
- The script tests the target availability (ping) before starting the installation
- The script assigns the account used for the connection based on the connection protocol, for example, smb → Administrator / rexec,ssh,rsh → root

Note: Before deploying the OS agent, always check the prerequisites on the target computer, such as the disk space. Most times, when the tacmd createNode command fails, the final error message available in the trace_cn.log indicates only the following:

“`An installation error occurred on host`**”.**

The script usage is as follows:

```
itso_inst_os_agent <ssh | rexec | smb> <target_hostname> <target_passwd>
<monitoring server> <backup monitoring server>
```

Example 2-9 shows the entire script used for the deployment.

Example 2-9 itso_inst_os_agent.cmd script listing

```
@echo off
::-------------------------------------------------------------
:: Product: itso_inst_os_agent.cmd
:: Description: simple script to deploy an OS agent to a
::    remote computer.
:: Created: Feb 23, 2006
:: Author : IBM Redbook Team
::-------------------------------------------------------------
::Port number to use for communications with the TEMS Remote (primary)
::1918 is the default
set P_PORT=2018
::Port number to use for communications with the TEMS Remote (secondary)
::1918 is the default
set S_PORT=2018
::Parameters
:: Connection protocol: ssh,rexec,rsh or smb
:: Note: the script assigns the account used for
:: the connection based on the connection protocol
:: e.g: smb -> Administrator / rexec,ssh,rsh -> root
set AccessType=%1
:: Target computer where OS agent is deployed
set Target=%2
:: Password for the remote account
set Passwd=%3
:: Monitoring Server hostname
set TEMS=%4
:: Backup Monitoring server hostname
set BTEMS=%5
::No parameter
if [%1]==[] goto HelpMsg
::Verify if the target is up
ping %Target% -n 1 -w 3000 | find "TTL="
if errorlevel 1 goto TargetDownMsg
::Assign the parameters (account,...) based on the access type
if %AccessType%==ssh goto UnixEnv
if %AccessType%==rexec goto UnixEnv
if %AccessType%==rsh goto UnixEnv
if %AccessType%==smb goto WinEnv
goto WrongAccessTypeMsg
::Define the parameters to access the Unix environment
:UnixEnv
set RemoteDir=/opt/IBM/ITM
set Account=root
goto InstallAgent
::Define the parameters to access the Windows environment
:WinEnv
set RemoteDir=c:\IBM\ITM
set Account=Administrator
goto InstallAgent
:InstallAgent
echo on
```

```
tacmd createNode -h %AccessType%://%Target% -u %Account% -w %Passwd% -d
%RemoteDir% -p PROTOCOL1=IP.PIPE PROTOCOL2=IP.UDP PORT=%P_PORT% SERVER=%TEMS%
BACKUP=YES BSERVER=%BTEMS% BPROTOCOL1=IP.PIPE BPROTOCOL2=IP.UDP PORT=%S_PORT%
@echo off
goto End
:HelpMsg
echo Usage: itso_inst_os_agent ^<ssh^|rexec^|smb^> ^<target_hostname^>
^<target_passwd^> ^<monitoring server^> ^<backup monitoring server^>
goto End
:TargetDownMsg
echo Error: %Target% is not reachable
goto End
:WrongAccessTypeMsg
echo Error: Access type -%AccessType%- is unknown
goto End
:End
```

2.2.11 An alternative solution

The other solution has been tested successfully with an environment based on a Tivoli Enterprise Monitoring Server Hub running on Windows 2003 and with z/OS Tivoli Enterprise Monitoring Servers Remote connected to the Hub. The basic installation process of the Tivoli Enterprise Monitoring Server Hub does not require any specific process other than those described in *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

IBM Tivoli Enterprise Monitoring Server Hub

This section describes the miscellaneous tasks involved in customizing the Tivoli Enterprise Monitoring Server Hub in order to address the requirements of our environment. Following are the tasks:

- Defining the network interface to communicate with Tivoli Enterprise Monitoring Server Remote

  After completing the basic installation, because the Tivoli Enterprise Monitoring Server Hub has several NICs, it is necessary to specify the one used by IBM Tivoli Monitoring. Refer to “Multiple network interface cards” on page 89 for details.

- Adding the application support for z/OS on the Tivoli Enterprise Monitoring Server Hub

  After completing the installation of the Tivoli Enterprise Monitoring Server Hub, in order to be able to manage the agents running on z/OS, run the application support as described in “IBM Tivoli Enterprise Monitoring Server Hub” on page 107. However, specify only the task of installing the application support (catalog and attribute files) on the Tivoli Enterprise Monitoring Server Hub, as shown Figure 2-46.

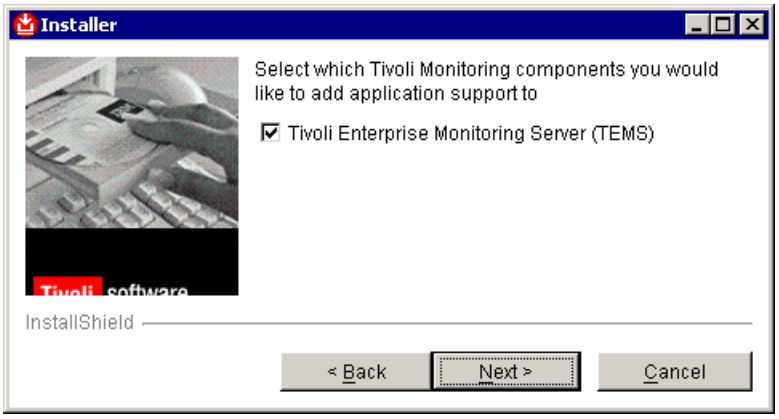

Installer
Select which Tivoli Monitoring components you would like to add application support to
Tivoli Enterprise Monitoring Server (TEMS)
InstallShield
< Back
Next >
Cancel

Figure 2-46 Application support for z/OS agents on Windows Tivoli Enterprise Monitoring Server Hub

2.2.12 Installing the key components for the distributed environment

This section describes the installation of the key components for the distributed environment.

Tivoli OMEGAMON XE for WebSphere Application Server

The Smart Bank environment has one standalone WebSphere V5.0 server that is used to run the Siebel Analytics program and two groups (cells), with the WebSphere V6.0 servers managed from two WebSphere Application Server Network Deployment.

The WebSphere Application Server V6.0 are installed on the following platforms.

- SUSE Linux Enterprise Server 9 Intel
- SUSE Linux Enterprise Server 9 for IBM eServer zSeries 64-bit

The WebSphere Application Server V5.0 is installed on the AIX 5.2 64-bit platform

The monitor for WebSphere does not yet have a package built and released for IBM Tivoli Monitoring V6.1. (This support is planned for 2006; check with your local IBM Sales Representative for the latest update.) However, the Tivoli OMEGAMON XE agent can interact with the IBM Tivoli Monitoring environment.

This chapter focuses on the installation of the Tivoli OMEGAMON XE agent in the WebSphere V6.0 environment. However, the explanations are applicable to the WebSphere V5.0 environment as well.

Before starting the installation of the agent, ensure the following:

- Download the Fix Pack 1.3.0-TIV-OXEWebSphere Application Server-IF0005 to support the WebSphere Application Server V6.0. This Fix Pack is applied after the installation of the agent code.
- Ensure that the application support for Tivoli OMEGAMON XE for WebSphere Application Server is installed on all the Tivoli Enterprise Monitoring Servers and Tivoli Enterprise Portal Servers.

Installation

The installation on z/OS fails with the following error message:

```
Memory fault install.sh failure:  terminating ... license declined.
```

As a workaround, before launching the installation script (install.sh), specify an alternate Java run-time environment (JRE™) path to Java 1.4.2 (Linux on IBM System z 64-bit). In this case, we use the Java available in the IBM Tivoli Monitoring OS agent directory.

From the ksh shell, enter the following command:

```
export alternateJRE=/opt/IBM/ITM/JRE/ls3266
```

When executing the install.sh script, ensure the following:

- The OMEGAMON agent home directory (CANDLEHOME) is /opt/IBM/Omegamon. This may be different in your environment. This directory must be separate from the one used for the IBM Tivoli Monitoring agents.
- Choose the WebSphere Application Server Monitoring Agent V130R205.

Configuration

The configuration program guides you in defining the protocols and ports used for the connection to the Tivoli Enterprise Monitoring Servers.

- Due to constraints in our architecture, only the following are specified:
  - A secondary Tivoli Enterprise Monitoring Server (Candle Management Server in OMEGAMON)
  - Port 2018 for communicating with Tivoli Enterprise Monitoring Servers
  - When the server has multiple network cards, the Primary Network Name for communicating with other IBM Tivoli Monitoring components updates the KDCB0_HOSTNAME variable.
- You must know the directory where WebSphere is installed.
- To configure the agent, execute the following command:

  ```
  <CANDLEHOME>/CandleConfig -A we
  ```

Example 2-10 shows the output of the configuration program on Linux.

Example 2-10 Configuration of the Tivoli OMEGAMON XE Agent for WebSphere Application Server

```
./CandleConfig -A we
CandleConfig        :  installer level  350 / 580.
CandleConfig        :  running li624 jre.

Will this agent connect to a CMS? [YES or NO] (Default is: YES): YES
CMS Host Name (Default is: blade3): blade7.production.local

Will the agent connect through a firewall? [YES or NO] (Default is: NO):

Network Protocol [ip, sna, or ip.pipe] (Default is: ip): ip.pipe

     Now choose the next protocol from one of these:
     - ip
     - sna
     - none
Network Protocol 2 (Default is: none): ip
```

```
     Now choose the next protocol from one of these:
     - sna
     - none
Network Protocol 3 (Default is: none):

IP Port Number (Default is: 1918):2018
IP.PIPE Port Number (Default is: 1918):2018
Enter name of KDC_PARTITION (Default is: null):

Configure connection for a secondary CMS? [YES or NO] (Default is: NO): YES
Secondary CMS HostName (Default is: blade3): blade11.production.local

Will the agent connect through a firewall? [YES or NO] (Default is: NO):

Secondary CMS protocol [ip, sna, or ip.pipe] (Default is: ip): ip.pipe

     Now choose the next protocol from one of these:
     - ip
     - sna
     - none
Secondary CMS Protocol 2 (Default is: none): ip

     Now choose the next protocol from one of these:
     - sna
     - none
Secondary CMS Protocol 3 (Default is: none):
Secondary CMS IP Port Number (Default is: 1918):2018
Secondary CMS IP.PIPE Port Number (Default is: 1918):2018
Enter Optional Primary Network Name or "none" (Default is: none): 10.1.1.126
WebSphere path(Default is: /opt/WebSphere/AppServer): /opt/IBM/WebSphere/AppServer
kwe.xml updated
Agent configuration completed...
```

Applying the Fix Pack

Perform the following tasks to apply and configure the Fix Pack on the Tivoli OMEGAMON agent:

1. Stop the Tivoli OMEGAMON agent using the following command:

   ```
   <CANDLEHOME>/bin/CandleAgent stop we
   ```

2. To stop all the monitored application servers:

   a. First retrieve the running servers, using the following command:

      ```
      <WAS_dir>/bin/serverStatus.sh -all
      ```

   b. Then use the following command

      ```
      <WAS_dir>/stopServer.sh <application_server_name>
      ```

3. At a shell prompt, from the <CANDLEHOME> directory, run the following commands to extract the patch for the Linux or the AIX environment:

   – For Linux Intel, run the following command:

     ```
     tar xvf <patch_dir>\weli4005.tar
     ```

   – For Linux on IBM System z, run the following command:

     ```
     tar xvf <patch_dir>\wels4005.tar
     ```

– For AIX, run the following command:

```
tar xvf <patch_dir>\weaix4005.tar
```

4. Run the class loader setup script:

```
<CANDLEHOME>/bin/setupCandleWAEnabler.sh
```

5. Restart the application servers that were stopped in step 2 on page 110.

```
<WebSphere Application Server_DIR>/bin/startServer.sh <server_name>
```

6. Update the <CANDLEHOME>\config\kwe.xml file.

   In order to support WebSphere V6.0, the agent's configuration file (kwe.xml) must be set up as follows:

   – The parameter WebSphere Application ServerAppServerRoot indicates the directory path name to the location where the WebSphere Application Server product is installed

   – A new parameter called WebSphere Application ServerInstanceRoot indicates the directory path name to the profile to be monitored by the agent

   Note: WebSphere V6.0 contains at least two separated environments. The first one contains only the binary files. The others (called profiles) contain a run-time execution environment that includes configuration files, the default location for the deployed applications, logs, and other data. All the profiles on a machine can share the same binary files.

   Example 2-11 shows a partial output of the kwe.xml file with the variables that you require to initialize, based on your WebSphere V6.0 environment.

Example 2-11 kwe.xml partial output

```
<?xml version="1.0" encoding="UTF-8"?>
<KWEAGENT VERSION="130"
   AgentID="AppSrv41"
    ...
   WebSphere Application
ServerInstanceRoot="/opt/IBM/WebSphere/AppServer/profiles/AppSrv41"
   WebSphere Application ServerAppServerRoot="/opt/IBM/WebSphere/AppServer"
  >
</KWEAGENT>
```

7. Restart the agent with the following command:

```
<CANDLEHOME>/CandleAgent start we
```

8. View the log files located in the directory <CANDLEHOME>/logs in order to verify whether the agent has started well without any error.

Setting instrumentation levels to collect Performance Monitoring Infrastructure data

The Performance Monitoring Infrastructure (PMI) application programming interface (API) extracts WebSphere Application Server performance data.

The following Tivoli Enterprise Portal workspaces require the PMI:

- DB connection pools
- Web applications
- Servlet/JavaServer™ Pages (JSP™)

- Servlet/JSPs for the selected Web application
- Enterprise JavaBeans™ (EJB™) containers
- Container transactions
- Container object pools
- Enterprise JavaBeans
- Enterprise JavaBeans methods for the selected bean
- Java 2 Platform, Enterprise Edition (J2EE) connector connection pools
- Servlet sessions
- Dynamic cache
- Workload management
- Thread pools
- Application server (some of the columns)

Perform the following tasks to set the instrumentation levels in order to collect the performance data on IBM WebSphere V6.0:

1. Open the URL `http://<was_server>:9060/admin`
2. Enter your account.
3. In the left frame, select **Servers**.
4. Select **Application Servers**.
5. In the right frame, choose the application server whose configuration you want to modify.
6. Select an initial specification level that is higher than None for Performance Monitoring Infrastructure.
7. Click the **Apply** button.
8. To save the update, click the **Save** hyperlink.

Preparing the instrumentation control file for workloads

An agent can be configured to collect the workload analysis data, including statistics about heap usage and garbage collections.

The following Tivoli Enterprise Portal workspaces require the collection of workload analysis data:

- All workloads
- Selected workload delays
- Longest running workloads
- Data sources
- HTTP sessions
- JMS summary
- In-flight workloads

To implement this instrumentation, perform the following tasks:

1. Edit the file kweiuser.xml located in the <CANDLEHOME>\config directory.

   Note: For a production environment, where a minimum overhead is expected, review carefully the parameters used for the instrumentation.

2. Run the Candle class loader setup script:

   `<CANDLEHOME>/bin/setupCandleWAEnabler.sh`

3. Open the following URL:

   `http://<was_server>:9060/admin`

4. Enter your account.
5. In the left frame, select **Servers**.
6. Select **Application Servers**.
7. In the right frame, select the application server whose configuration you want to modify.
8. Select **Java** and **Process Management**.
9. Select **Process Definition**.
10. Select **Java Virtual Machine**.
11. In the Boot Class Path box, enter the following:

    ```
    <CANDLEHOME>/classes/kwescl.1.4.2.jar:<CANDLEHOME>/classes/kwewa.jar
    Caution: <CANDLEHOME> is the installation directory of the Omegamon Agent
    ```

12. In the Generic JVM Arguments box, add the following:

    ```
    -Dcom.candle.kwe.instrument.agent_parms=<CANDLEHOME>/config/kweiuser.xml
    ```

13. Click **Apply**.
14. Click **Save**.
15. From the Tivoli Enterprise Portal, in the Physical View, right-click your application server and select **Take Action** from the pop-up menu.
16. In the Take Action window, select the WebSphere Application Server Configuration Setup command and enter the following parameters:

    ```
    Workload_Analysis_Y_or_N=Y
    VerboseGC_Y_or_N=Y (to collect garbage
    File=kweiuser.xml
    ```

17. Select the WebSphere Application Servers in the Destination Systems List, and click **OK**.
18. From the WebSphere Application Server Admin Console, recyle the application server.

    After executing all these tasks, you must have a view that is similar to that of the Tivoli Enterprise Portal after selecting the **All Workload** workspace. Figure 2-47 shows a window displaying the Tivoli Enterprise Portal output.

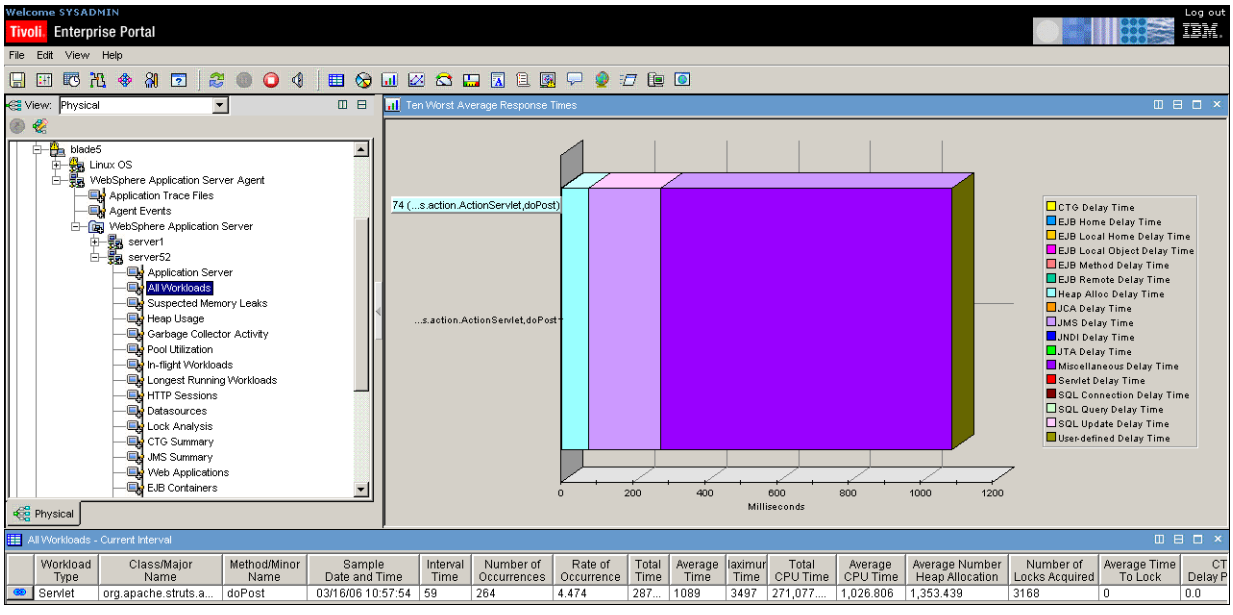


Figure 2-47 Tivoli Enterprise Portal: WebSphere All Workload workspace

Using the Universal Agent to monitor the IBM HTTP Server

The IBM Tivoli Open Process Automation Library provides a package for monitoring the IBM HTTP V6.0 Web Server using the IBM Tivoli Monitoring Universal Agent. The IBM Tivoli Open Process Automation Library Web site is available on the Web at:

http://www-18.lotus.com/wps/portal/tm

The Universal Agent uses the file data provider to extract useful metrics from the access_log file regarding the health of the HTTP Server. Thus, data about the performance characteristics of the IBM HTTP V6.0 Web Server is available, including the following:

- The throughput of the Web server
- The number of Web hits per hour per day
- Errors, including the client that initiated the request to the Web server

The IBM HTTP V6.0 Web Servers are installed on two types of platforms:

- SUSE Linux Enterprise Server 9 Intel
- SUSE Linux Enterprise Server 9 for IBM eServer zSeries 64-bit

Log format

The IBM HTTP Server is based on Apache. The file data provider extracts useful data from the access_log file based on the parameters defined in the meta file (IBMHTTP.mdl). The default log format output defined in the HTTP server configuration file (httpd.conf) does not provide all the fields that match the meta file's attributes. However, the log file output can be customized to suit individual requirements.

The httpd.conf contains directives relating to the log file format:

- The LogFormat directive defines which information must be redirected to a log file. The use of LogFormat is described in the following Web site:

  http://httpd.apache.org/docs/2.0/mod/mod_log_config.html#logformat

  Several LogFormat directives can be defined.
- The CustomLog directive defines the log file name and the LogFormat used to define the log format output. Table 2-6 shows the correspondence between meta file attributes and LogFormat strings.

Table 2-6 Meta file attributes and LogFormat strings

Meta file attribute	LogFormat string
ClientLocation	%h
ClientUserName	%l
Authorized_User	%u
Date_Time	%t
Time_Zone	%t
Request	%r
ServiceStatus	%s
BytesReceived	%I (requires mod_logio module otherwise put hard-coded value 0)
Referral	%{Referer}i -> %U
Browser	%{User-Agent}i

Meta file attribute	LogFormat string
Service	- (not implemented)
ServerName	%v
RequestParameters	%q
BytesSend	%O (requires mod_logio module otherwise use %b)
RequestElapsedTime	%T

Note: The module mod_logio is supported since Apache V2.0.43 was introduced. The IBM HTTP Server V6 is based on Apache 2.0.47, but does not include this module. However, a third-party module can be used. This solution has *not* been fully tested in our project.

If the mod_logio module is not implemented, there is no way of feeding the BytesReceived attribute. Therefore, the %I must be replaced with the hardcoded value 0. The BytesSend attribute must get the value returned by %b instead of %O. However, the value returned by %b and %O are not identical. The %b format string represents only the size of the HTTP response in bytes and not the actual number of bytes sent over the network.

- LogFormat directive with the mod_logio module enabled

  Example 2-12 shows the definition of the LogFormat directive called *itmuniagent* (nickname) if the mod_logio module is enabled. By default, the CustomLog directive for the access_log file uses the nickname *common*. To change the log format output matching the attributes of the Universal Agent meta file, the CustomLog directive must use the nickname itmuniagent instead of common.

Example 2-12 HTTP Server: LogFormat and CustomLog directives with the mod_logio module

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%h %l %u %t \"%r\" %>s %I \"%{Referer}i -> %U\" \"%{User-Agent}i\" - %v
%q %O %T" itmuniagent

#Do not use the common LogFormat
#CustomLog logs/access_log common
#ITM Universal Agent log format
CustomLog logs/access_log itmuniagent
```

- LogFormat directive without the mod_logio module

  Example 2-13 shows the definition of the LogFormat directive called *itmuniagent* (nickname) without the mod_logio module.

Example 2-13 HTTP Server: LogFormat and CustomLog directives without the mod_logio module

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%h %l %u %t \"%r\" %>s 0 \"%{Referer}i -> %U\" \"%{User-Agent}i\" - %v
%q %b %T"
itmuniagent
```

```
#Do not use the common LogFormat
#CustomLog logs/access_log common
#ITM Universal Agent log format
CustomLog logs/access_log itmuniagent
```

Deploying the Universal Agent

The OS agents were deployed earlier on all our systems. Therefore, deploying other agents from the Windows Tivoli Enterprise Monitoring Server Remote is possible. (This must also be carried out from the AIX Tivoli Enterprise Monitoring Server Remote.)

In order to quickly deploy the Universal Agent, initiate the installation from the command prompt by performing the following tasks:

1. Place the IBMHTTP.mdl file in the directory <ITM_install dir>\CMS\depot\UACONFIG.
2. Edit the IBMHTTP.mdl file and verify whether the path defined in the //SOURCE control statement matches the location of the access_log file on the target computer.
3. Deploy the Universal Agent on the Linux remote target with the following command:

   ```
   tacmd addsystem -t um -n <hostname>:LZ -p UA.CONFIG="IBMHTTP.mdl"
   ```

Note: After the first installation of the Universal Agent, the TEP is unable to display the contents of the Universal Agent workspaces. After connecting the Universal Agent to a Tivoli Enterprise Monitoring Server Remote, CAT and ATTR files are not sent to the Tivoli Enterprise Monitoring Server Hub, and Object Description Identifier (ODI) files are not sent to the Tivoli Enterprise Portal Server. Therefore, the Tivoli Enterprise Portal workspaces are unable to show the data.

In order to work around this problem, connect the Universal Agent to the Tivoli Enterprise Monitoring Server Hub, where the MDL file is being imported. After the Model File Element (MDL) file is imported, the Tivoli Enterprise Portal workspaces show data. Thus, the Universal Agent can move to any remote Tivoli Enterprise Monitoring Server. The reason is that the CAT and ATTR files are created for Tivoli Enterprise Monitoring Server and an ODI file is created on the TEP by the Tivoli Enterprise Monitoring Server Hub, which is the only one with the ability to do so.

Simple network monitoring with the Universal Agent

A simple solution must be implemented to monitor the production network in order to periodically check the availability of the Smart Bank servers.

The Simple Network Management Protocol (SNMP) data provider available in the Universal Agent is a full function of the SNMP manager, which is capable of discovering the network.

Network monitoring must address the following requirements:

- Only the production network must be discovered.
- A workspace must display only the network reachability of the participating servers to the Smart Bank environment.

Simple Network Management Protocol data provider setup

The basic installation of the Universal Agent is done on a Windows platform. (It must be done on UNIX as well.)

After completing the installation of the Universal Agent, customize it by performing the following tasks:

1. Update the Universal Agent's configuration file. Table 2-7 shows the basic parameter settings of the Universal Agent's configuration file (KUMENV).

Table 2-7 SNMP data provider settings

Parameter setting	Description
KUMA_STARTUP_DP=SNMP	The Universal Agent starts the SNMP data provider during the startup
KUMP_SNMP_NET_DISCOVERY=Y	All the network resources are discovered (servers, gateways, and so on)
KUMP_SNMP_MANAGE_LOCAL_NETWORK=Y	Manage the local network
KUMP_SNMP_NET_DISCOVER_ENTERPRISE=N	Do not manage the status of the entire network enterprise

Example 2-14 shows a partial output of the configuration file with the definition of the parameters relating to the SNMP Data Provider. The Universal Agent runs on a server with multiple network cards. Therefore, the KDCB0_HOSTNAME parameter is initialized.

Example 2-14 Partial listing of the KUMENV file

```
*-------------------------------------------------------------------*
* Needed for multi-home system (more than one network interface)  *
*-------------------------------------------------------------------*
KDCB0_HOSTNAME=10.1.1.131
* KUM_DCH_HOSTNAME=
* KUM_DP_HOSTNAME=
* KUM_DCH_HOST=
* KUMP_DCH_HOST=
*-------------------------------------------------------------------*
* UA Startup automatic start DP options                            *
* (ASFS,APIS,FILE,SOCK,HTTP,SNMP,POST,ODBC,SCRP)                   *
*-------------------------------------------------------------------*
KUMA_STARTUP_DP=SNMP
...

*-------------------------------------------------------------------*
* SNMP DP Parameters                                               *
*-------------------------------------------------------------------*
KUMP_SNMP_MONITOR_TRAP=N
KUMP_SNMP_NET_DISCOVERY=Y
KUMP_SNMP_NET_DISCOVER_ENTERPRISE=N
KUMP_SNMP_MANAGE_LOCAL_NETWORK=Y
KUMP_SNMP_NET_COMMUNITY=public
...
```

2. Specify a logical name for the production network by editing the symbolic name file KUMSNAME located in <ITM_install_dir>\tmaitm6\work. Example 2-15 shows the contents of this file.

Example 2-15 Contents of the KUMSNAME file

```
************************************************************************
* Network Symbolic Name Table                                          *
* Specify symbolic name of discovered networks                         *
* Format:                                                              *
*                                                                      *
* Network-address Symbolic-name-string                                 *
*                                                                      *
* For example:                                                         *
* 198.210.37.0 Westlake-CTDEV                                          *
* 10.10.18.0 DistSystemLab                                             *
************************************************************************
*Production network dedicated for the inter-servers communications
10.1.1.0 Production-Net
```

3. In the KUMSLIST file located in <ITM_install_dir>\tmaitm6\work, define the name of the file containing the “Host Lists” (managed node lists) of the Smart Bank servers. Add the following line in this file:

   ```
   kumlist_Smart Bank showcase.def
   ```

4. Create the Hot Lists file kumlist_Smart Bank showcase.def in the directory <ITM_install_dir>\tmaitm6\work, and specify the list of hosts.

   Example 2-16 shows a partial static list of hosts belonging to the Smart Bank environment. Refer to *IBM Tivoli Monitoring Agent User’s Guide*, SC32-9459 for details about defining the lists based on the resource filter.

Example 2-16 Partial listing of the kumslist_Smart Bank showcase.def

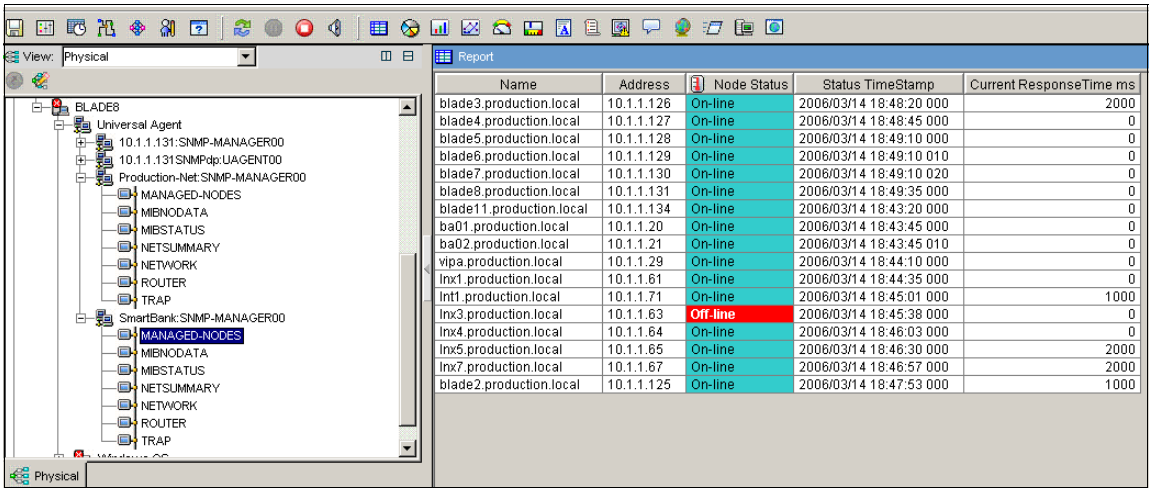
```
LISTNAME=Smart Bank showcase
blade2
blade3
blade4
...
lnt1
lnx3
lnx4
lnx5
lnx7
```

5. Recycle the Universal Agent.

6. Load the Hot Lists by issuing the following command:

   ```
   kumpcon loadlist kumslist_Smart Bank showcase.def
   ```

Figure 2-48 displays a window showing the TEP output. The default report is customized partially for a better understanding of the published data.



Name	Address	Node Status	Status TimeStamp	Current ResponseTime ms
blade3.production.local	10.1.1.126	On-line	2006/03/14 18:48:20 000	2000
blade4.production.local	10.1.1.127	On-line	2006/03/14 18:48:45 000	0
blade5.production.local	10.1.1.128	On-line	2006/03/14 18:49:10 000	0
blade6.production.local	10.1.1.129	On-line	2006/03/14 18:49:10 010	0
blade7.production.local	10.1.1.130	On-line	2006/03/14 18:49:10 020	0
blade8.production.local	10.1.1.131	On-line	2006/03/14 18:49:35 000	0
blade11.production.local	10.1.1.134	On-line	2006/03/14 18:43:20 000	0
ba01.production.local	10.1.1.20	On-line	2006/03/14 18:43:45 000	0
ba02.production.local	10.1.1.21	On-line	2006/03/14 18:43:45 010	0
vipa.production.local	10.1.1.29	On-line	2006/03/14 18:44:10 000	0
lnx1.production.local	10.1.1.61	On-line	2006/03/14 18:44:35 000	0
lnt1.production.local	10.1.1.71	On-line	2006/03/14 18:45:01 000	1000
lnx3.production.local	10.1.1.63	Off-line	2006/03/14 18:45:38 000	0
lnx4.production.local	10.1.1.64	On-line	2006/03/14 18:46:03 000	0
lnx5.production.local	10.1.1.65	On-line	2006/03/14 18:46:30 000	2000
lnx7.production.local	10.1.1.67	On-line	2006/03/14 18:46:57 000	2000
blade2.production.local	10.1.1.125	On-line	2006/03/14 18:47:53 000	1000

Figure 2-48 TEP: Monitoring the servers' availability

Monitoring the Uniform Resource Locators with the Universal Agent

A simple solution to monitor the availability and the response time of the Uniform Resource Locator (URL) requests to the Smart Bank Web servers must be implemented. The HTTP data provider available in the Universal Agent provides the ability to monitor URLs in order to diagnose the availability and response time of the remote Web servers. *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143 explains the configuration of the data provider in detail.

After installing the Universal Agent, perform the configuration as follows:

1. Update the parameter KUMA_STARTUP_DP in the Universal Agent's configuration file (KUMENV): KUMA_STARTUP_DP=HTTP.
2. Update the URL list in the KUMPURLS file located in the directory <ITM_install_dir>\tmaitm6\work, as shown in Example 2-17. The default polling of 300 seconds is changed to 120 seconds (StatusInterval parameter) for each HTTP link.

Example 2-17 Updating the URL list

```
* List of URLs to monitor by the Universal Agent HTTP Data Provider
blade2.production.local/fss StatusInterval=120
lnx1.production.local/fss StatusInterval=120
safss:9080/web/app StatusInterval=120
```

Note: In the URL list, it is not possible to specify the Web pages protected by user name and password at the application level. The HTTP data provider only tests the URL availability. It does not have the capability to manage interactive sessions. The product, IBM Tivoli Configuration Application Manager for Response Time Tracking, addresses this requirement.

3. Recycle the Universal Agent.

Note: After the installation of the Universal Agent, the TEP is unable to display the contents of the MANAGED_URL workspace. The following error is displayed:

```
KFWITM220E Request failed during execution
```

This error occurs when the Universal Agent is connected to a Tivoli Enterprise Monitoring Server Remote. To work around this, connect the Universal Agent directly to the Tivoli Enterprise Monitoring Server Hub. When the Universal Agent is successfully connected to the Tivoli Enterprise Monitoring Server Hub, the workspace is able to display data. Later, return to the original Tivoli Enterprise Monitoring Server Remote. The reason for this is described in “Using the Universal Agent to monitor the IBM HTTP Server” on page 114.

3

Using Physical views

This chapter describes the main workspaces that we used in the Smart Bank project. These workspaces are provided by different products:

- From IBM Tivoli Monitoring. This is described in 3.1, “Physical views from IBM Tivoli Monitoring V6” on page 122.
- From IBM Tivoli Composite Application Manager. This is described in 3.2, “Physical views from IBM Tivoli Composite Application Manager” on page 131.
- From IBM OMEGAMON XE for MQ. This is described in 3.3, “IBM CICS Business Event Publisher for MQ Series and event processing” on page 154.
- From the distributed environment. This is described in 3.4, “Multichannel architecture: Physical views” on page 170.

© Copyright IBM Corp. 2006. All rights reserved.

3.1 Physical views from IBM Tivoli Monitoring V6

With IBM Tivoli Monitoring, the Tivoli Enterprise Portal Client and Server application was introduced. This application enables user monitoring and easy access of all the required system and application performance information across the entire enterprise and across platforms.

As described in the introduction to IBM Tivoli Monitoring V6 product in “IBM Tivoli Monitoring V6” on page 24, the portal acts as the single point of control for all the activities around Tivoli monitoring. IBM Tivoli Monitoring V6 enables the following tasks:

- Checking the health of the monitoring infrastructure
- Getting all the monitoring information from a single user interface
- Creating and distributing the monitoring rules, that is, situations, to define specific thresholds that must lead to events
- Viewing events across the entire monitoring environment, including the Tivoli Enterprise Console
- Implementing predefined activities that are to be executed manually or automatically in order to manage critical issues in the IT environment
- Interoperating with system automation tools in order to control more complex automation tasks

By installing a specific agent on a system and connecting it to the IBM Tivoli Monitoring V6 infrastructure, an entry in the physical tree is created. Figure 3-1 shows the physical tree for the Smart Bank showcase.

For each system (operating system group) platform, an agent is installed, and an item entry is created. Below this level, the different physical or logical system names on which the agents are running on are visible. For each agent type, a subtree is created. This subtree contains attribute groups that are dependent on the type of the reporting agent.

If there are multiple instances, for example, on z/OS subsystems on the same system, each instance has its own subtree.

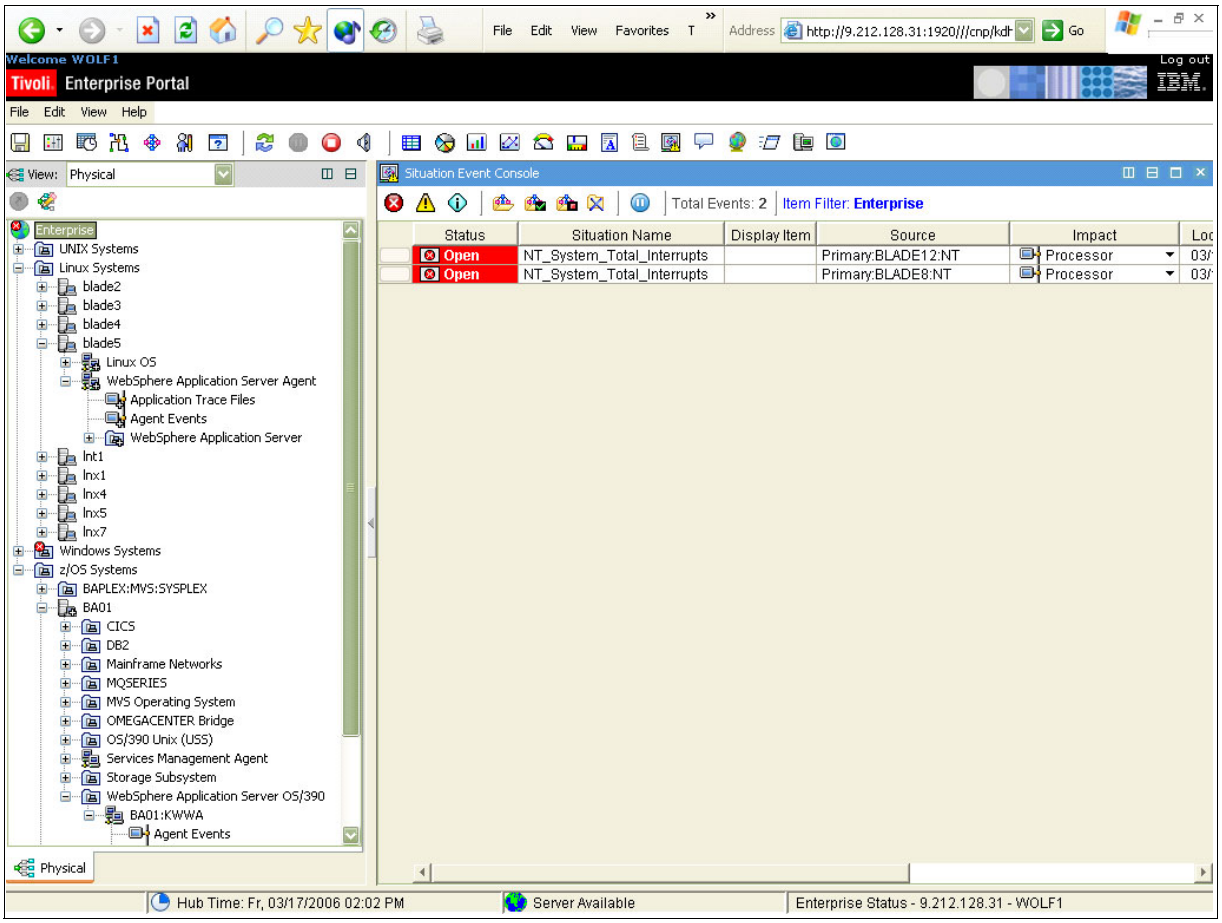


Figure 3-1 The physical tree for the Smart Bank showcase

Each entry in the tree is called a navigator item. Each navigator item has at least one workspace. By default, it is named as the navigator item entry itself.

A workspace may contain several views. Figure 3-2 shows an example from the Windows Server® navigator subtree. Multiple workspaces are available, with each of them detailing different values in different views on the process information gathered by the Windows Server agent. A single view may contain different methods of presentation, but only one can be used at a time. Following are the presentation types that are available:

- Circular Gauge
- Linear Gauge
- Bar Chart
- Table
- Pie Chart
- Plot Chart
- Notepad
- Situation Event Console
- Universal Message Console
- Graphic View
- Take Action
- Terminal
- Browser

We recommend that you consult the user manual about how to use the different options that are available.

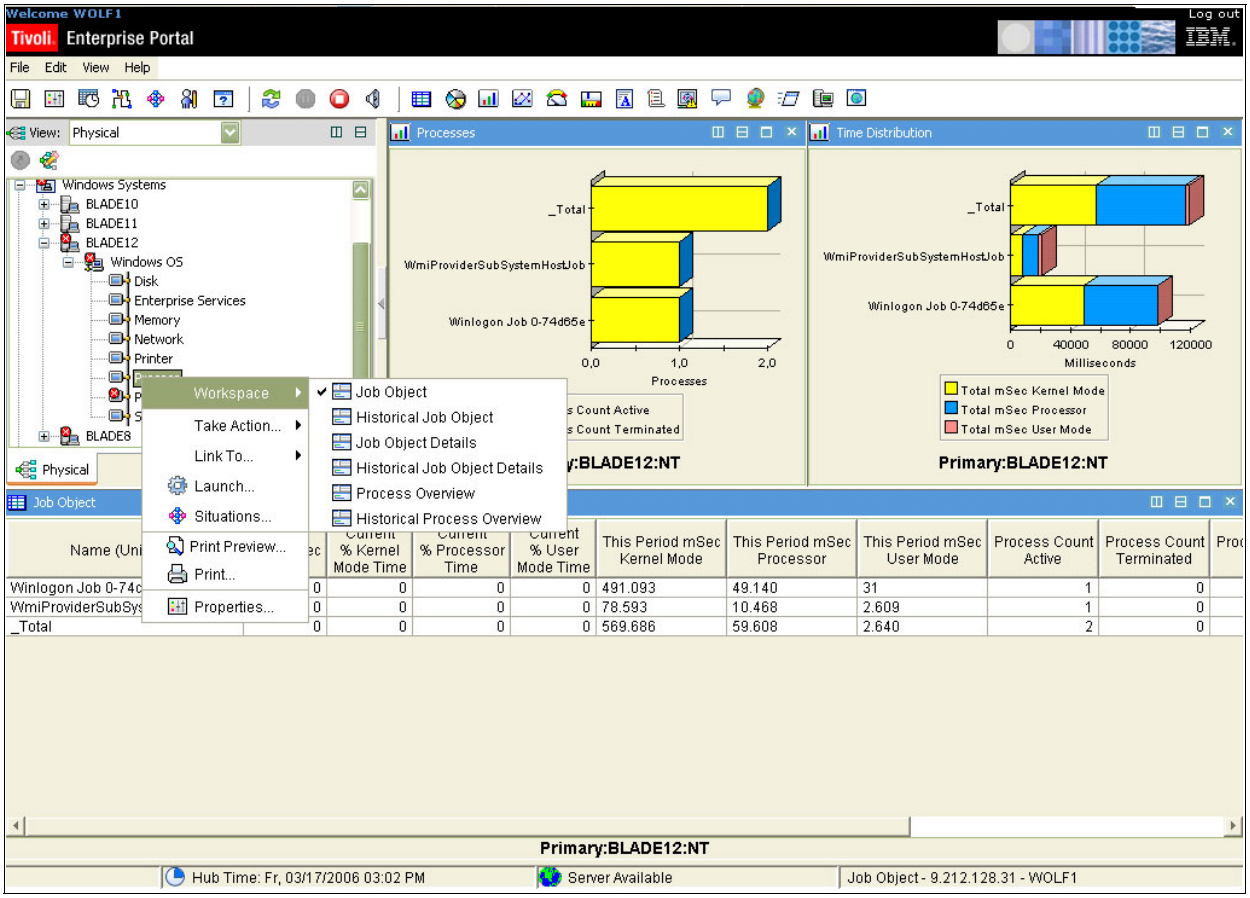


Figure 3-2 The Windows system navigator subtree: Process workspaces

In Figure 3-2, one Table view and two Bar Chart views are used. The different ways of presentation enable a quick overview of the key metrics presented.

A user can modify all the workspaces. However, the IBM Tivoli Monitoring V6 system administrator may prevent specific views from being changed. The user may also be forbidden from changing any of the workspaces. The IBM Tivoli Monitoring V6 system administrator decides on the rights the user gets. The system administrator creates its own views and makes it available to all the users for further use.

Following were the agents used for the Smart Bank showcase:

- IBM OMEGAMON for IBM z/OS V3.10
- IBM OMEGAMON for UNIX System Services (USS) V2.20
- IBM OMEGAMON for DB2 V3.10
- IBM OMEGAMON for Mainframe Networks V3.10
- IBM OMEGAMON for client Information Control System (CICS) V3.10
- IBM OMEGAMON for Storage on z/OS V3.1.0
- IBM OMEGAMON for WebSphere Application Server on z/OS V1.30 (part of IBM Tivoli Composite Application Manager)

- IBM OMEGAMON for WebSphere Application Server on Distributed System V1.30 (part of IBM Tivoli Composite Application Manager)
- IBM OMEGAMON for WebSphere Business Integration V1.10 (part of IBM Tivoli Composite Application Manager)
- IBM Tivoli Monitoring for Distributed Systems on Windows V6.10
- IBM Tivoli Monitoring for Distributed Systems on AIX V6.10
- IBM Tivoli Monitoring for Distributed Systems on Linux on IBM System x V6.10
- IBM Universal Agent V6.10
- IBM Tivoli Monitoring for Service-Oriented Architecture V6.0 (part of IBM Tivoli Composite Application Manager)

The following sections describe the various agents and answer the following key questions:

- What can be expected from this agent?

  Each product is represented by a specific agent called Tivoli Enterprise Monitoring Agent. The reported data is specific to the application.

- On which system must this specific agent type be installed?

  Not all the agents are installed on every system in the infrastructure. Agents are installed only to a specific system with a specific requirement.

- Which attributes does this agent support?

  Each agent type groups its gathered information into attribute groups.

- Which key points must be monitored from a system perspective?

  The different points of view within an IT organization may differ from team to team. The system group in charge of delivering a stable IT environment faces specific challenges.

- Which additional application must be covered by specific rules defined to this agent?

  Business unit managers who are in charge of a special part of the business are less interested in single system components because their main focus is the execution of their core business through the entire infrastructure. They require designated information about the health of their supporting applications.

3.1.1 IBM OMEGAMON for z/OS V3.10

This agent provides a deep insight into the z/OS operating environment. From V3.10, the sysplex monitoring functions are an integral part of the z/OS agent. Besides, the installation is simplified.

A navigation tree with detailed information about performance and configuration values inside the z/OS logical partitions (LPARs) and the sysplex components is provided to a user. Figure 3-3 shows the navigation tree from the z/OS agent. The agent presents a lot of information about the z/OS system and its major resources. This agent tracks the monitoring activities that help keep the system up and running, the health of the system, the existence of

the dedicated process, the correct usage of global resources, and all the other aspects regarding the z/OS operating system.

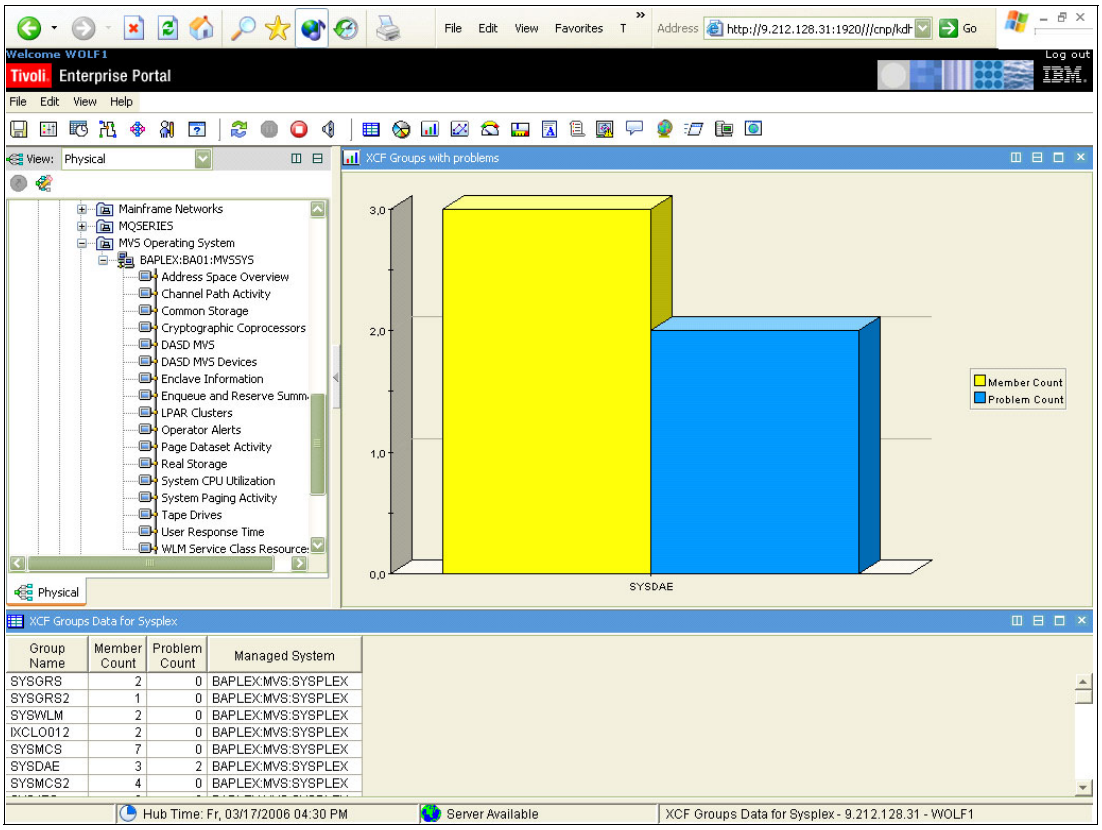


Figure 3-3 The z/OS navigation subtree

For the Smart Bank showcase, some applications have their own processes implemented directly in the operating system. If these processes are not performing well, the dedicated application fails. Most of the rules reflect the requirements of the system programmers, while a few rules reflect the requirements of the business managers. In a large environment, the system programmer or those in charge of the operations do not understand the impact of a missing task that is already started, and the impact of this on a company's business.

3.1.2 IBM OMEGAMON for UNIX System Services V2.20

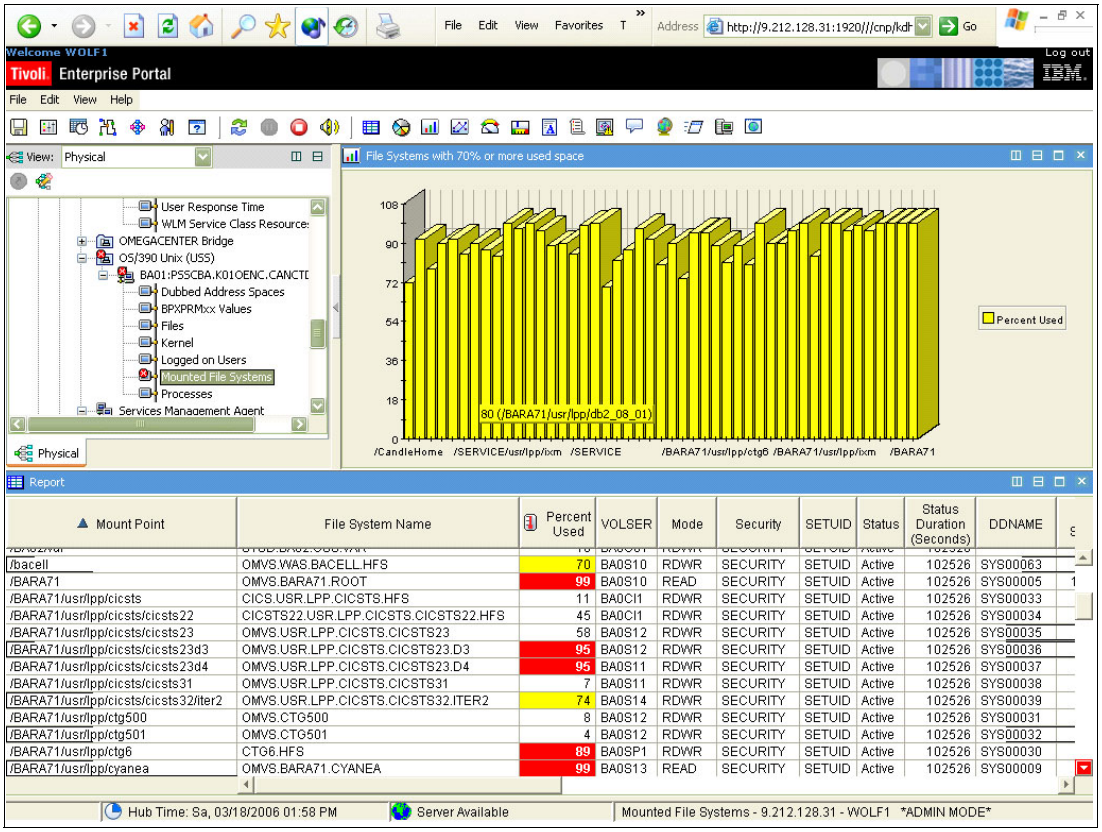

Welcome WOLF1
Tivoli Enterprise Portal
File Edit View Help
View: Physical
User Response Time
WLM Service Class Resource
OMEGACENTER Bridge
OS/390 Unix (USS)
BA01:PSSCBA.K01OENC.CANCTI
Dubbed Address Spaces
BPXPRMxx Values
Files
Kernel
Logged on Users
Mounted File Systems
Processes
Services Management Agent
Physical
File Systems with 70% or more used space
Percent Used
80 (/BARA71/usr/lpp/db2_08_01)
/CandleHome /SERVICE/usr/lpp/ixm /SERVICE /BARA71/usr/lpp/ctg6 /BARA71/usr/lpp/ixm /BARA71
Report
Mount Point | File System Name | Percent Used | VOLSER | Mode | Security | SETUID | Status | Status Duration (Seconds) | DDNAME
/bacell | OMVS.WAS.BACELL.HFS | 70 | BA0S10 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00063
/BARA71 | OMVS.BARA71.ROOT | 99 | BA0S10 | READ | SECURITY | SETUID | Active | 102526 | SYS00005
/BARA71/usr/lpp/cicsts | CICS.USR.LPP.CICSTS.HFS | 11 | BA0CI1 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00033
/BARA71/usr/lpp/cicsts/cicsts22 | CICSTS22.USR.LPP.CICSTS.CICSTS22.HFS | 45 | BA0CI1 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00034
/BARA71/usr/lpp/cicsts/cicsts23 | OMVS.USR.LPP.CICSTS.CICSTS23 | 58 | BA0S12 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00035
/BARA71/usr/lpp/cicsts/cicsts23d3 | OMVS.USR.LPP.CICSTS.CICSTS23.D3 | 95 | BA0S12 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00036
/BARA71/usr/lpp/cicsts/cicsts23d4 | OMVS.USR.LPP.CICSTS.CICSTS23.D4 | 95 | BA0S11 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00037
/BARA71/usr/lpp/cicsts/cicsts31 | OMVS.USR.LPP.CICSTS.CICSTS31 | 7 | BA0S11 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00038
/BARA71/usr/lpp/cicsts/cicsts32/iter2 | OMVS.USR.LPP.CICSTS.CICSTS32.ITER2 | 74 | BA0S14 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00039
/BARA71/usr/lpp/ctg500 | OMVS.CTG500 | 8 | BA0S12 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00031
/BARA71/usr/lpp/ctg501 | OMVS.CTG501 | 4 | BA0S12 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00032
/BARA71/usr/lpp/ctg6 | CTG6.HFS | 89 | BA0SP1 | RDWR | SECURITY | SETUID | Active | 102526 | SYS00030
/BARA71/usr/lpp/cyanea | OMVS.BARA71.CYANEA | 99 | BA0S13 | READ | SECURITY | SETUID | Active | 102526 | SYS00009
Hub Time: Sa, 03/18/2006 01:58 PM
Server Available
Mounted File Systems - 9.212.128.31 - WOLF1 *ADMIN MODE*

Figure 3-4 The physical tree of OMEGAMON for USS

OMEGAMON for USS provides a deep insight into the UNIX System Services running on z/OS. Because the implementation of the WebSphere Application Server and the TCP/IP stack are highly dependent on a healthy USS subsystem under z/OS, this monitor helps keep the system up and running.

Figure 3-4 shows that many mount points are affected by lack of free space, which may lead to a major system outage.

For the Smart Bank showcase, several situations are created to keep the file system clean in order to have all the required mount points up and running along with all the required USS processes in the system. Later, these rules are used partly, in order to set up the Application Logical views.

3.1.3 IBM OMEGAMON for DB2 V3.10

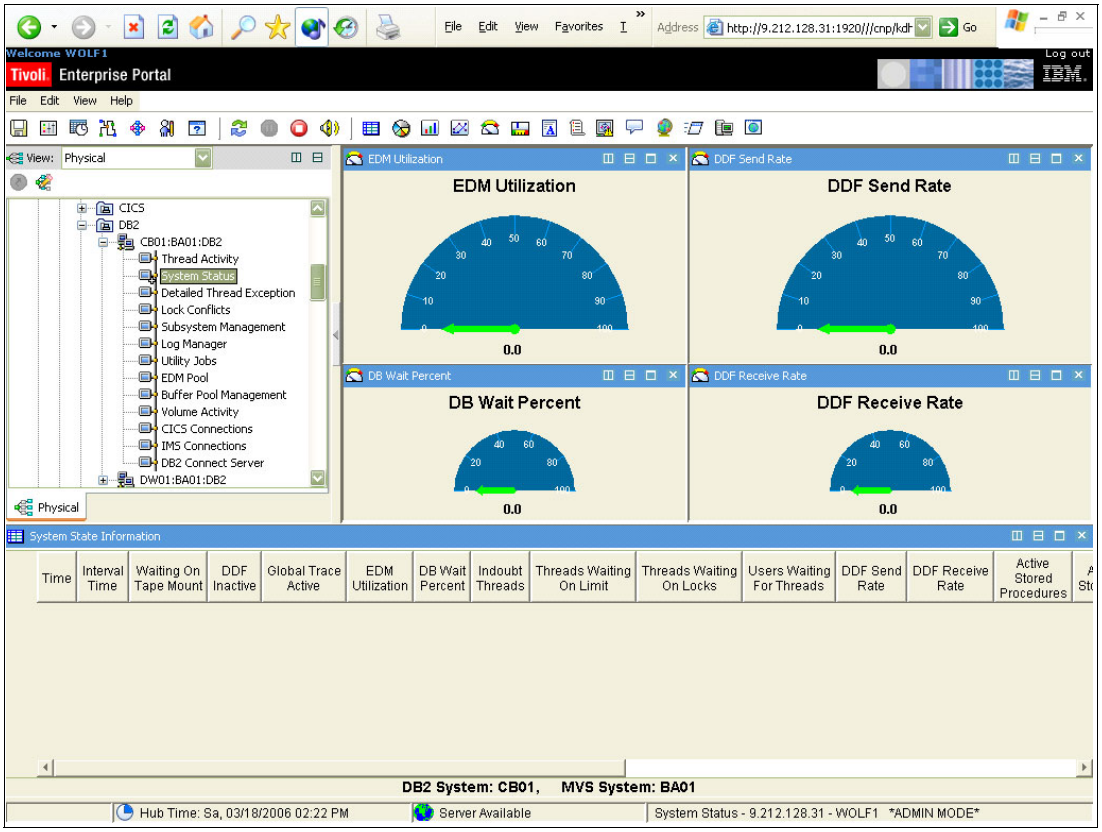


Figure 3-5 The navigation subtree of IBM OMEGAMON for DB2

OMEGAMON for DB2 offers plenty of information on the DB2 subsystem running on z/OS. Figure 3-5 shows the navigation tree generated by the IBM Tivoli Monitoring V6 system. If multiple DB2 subsystems are running on the same system, as in the case of the Smart Bank showcase's system, a tree is maintained for each subsystem.

The performance values provided are the key indicators of a healthy DB2 subsystem. Most of these values address the requirements of the database administrators and system programmers to successfully operate a DB2 subsystem.

In the Smart Bank showcase, the rules are defined to figure out if a dedicated application is in trouble or not. Although the used CICS systems are visible in the CICS connections workspace, if they have slow performing threads in DB2 or are creating a lot of lock conflicts, they are visible in the lock conflicts workspace.

3.1.4 IBM OMEGAMON for Mainframe Networks V3.10

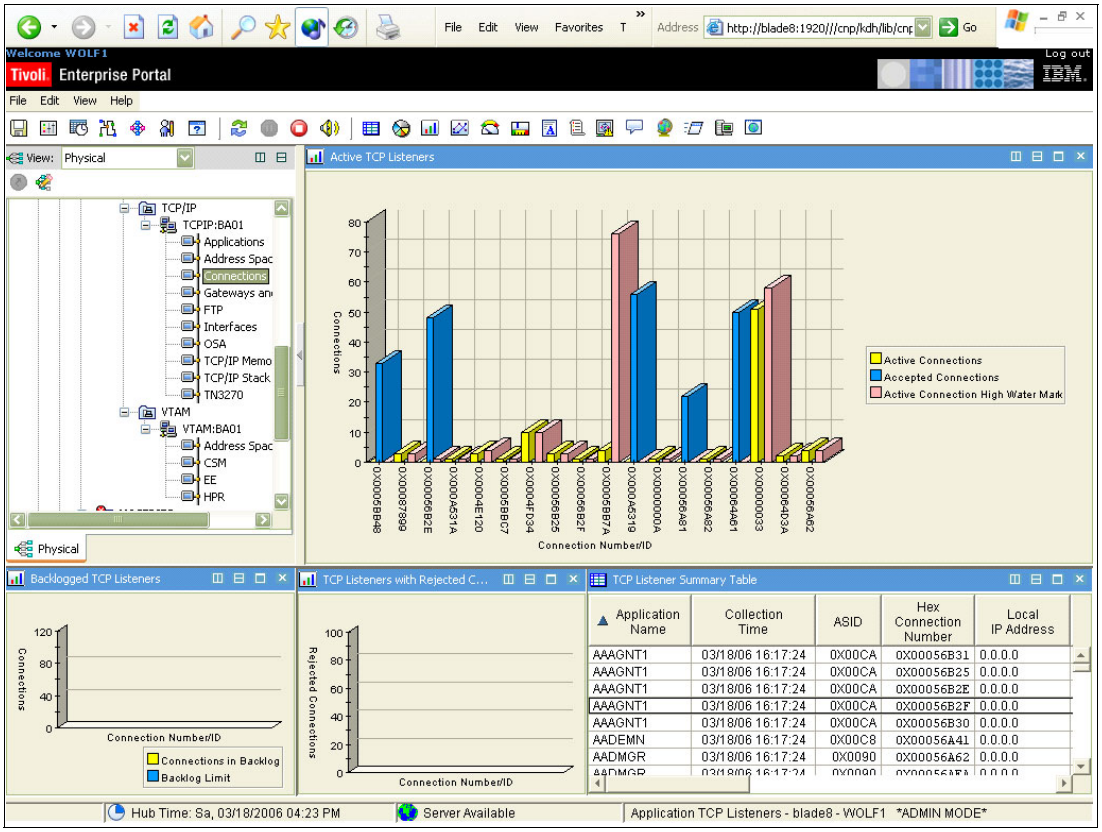


Figure 3-6 The navigation subtree of IBM OMEGAMON for Mainframe Networks

The navigation tree for the OMEGAMON for Mainframe Networks shown in Figure 3-6 includes the TCP/IP statistics and the Virtual Telecommunications Access Method (VTAM) attribute groups.

This is a system-focused monitor, providing a lot of performance statistics from the system, although many values can be used to observe how the application is progressing. This monitoring allows you to check specific ports and attached tasks so that the link between an open port and the according application may be tracked. After this, look at the link usage and understand how much network traffic is handled across this special connection, and whether any delays are reported.

3.1.5 IBM OMEGAMON for CICS V3.10

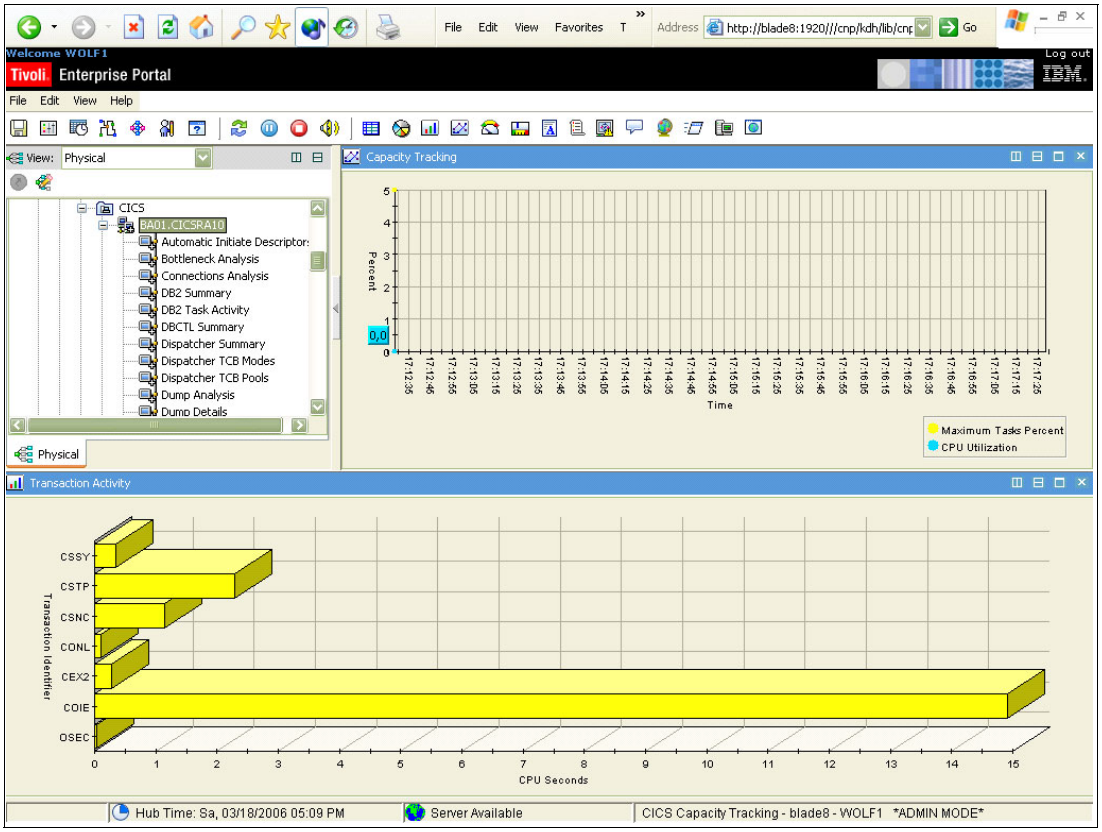


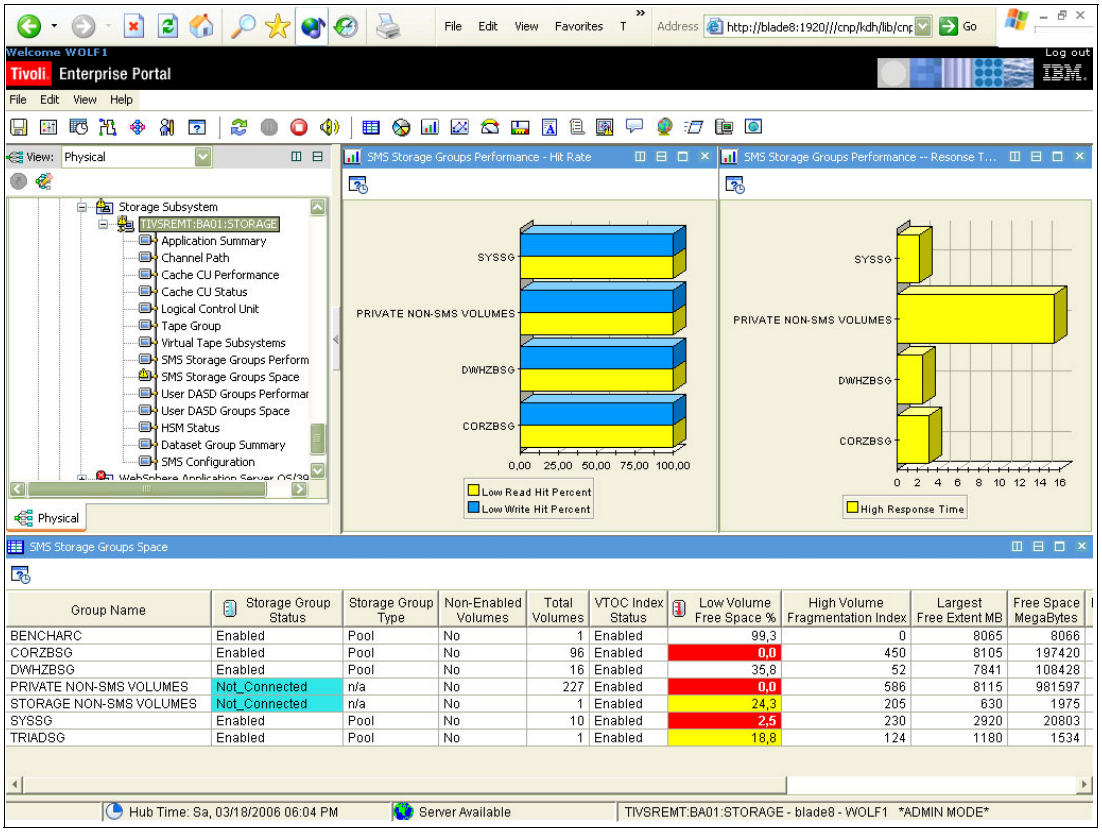
Figure 3-7 The navigation subtree of OMEGAMON for CICS

For each CICS region, a subtree with detailed attribute groups is generated in the physical tree of IBM Tivoli Monitoring V6, as shown in Figure 3-7.

New capabilities in the OMEGAMON XE for CICS on z/OS V3.1 release allow you to view extensive information about your CICS resources, for example, TCP/IP, storage use, and related Java applications, without looking at outside batch reports or information from other monitors or sources. The addition of more than 40 new tables and related attributes provides more detailed information to further reduce the time you spend pin-pointing problems affecting performance.

This monitor addresses the requirements to support system programmers and the operations staff, who are in charge of running the CICS system, as well as the requirements of the application department to get insight into the performance of the application.

3.1.6 IBM OMEGAMON for Storage on z/OS V3.1.0



Group Name	Storage Group Status	Storage Group Type	Non-Enabled Volumes	Total Volumes	VTOC Index Status	Low Volume Free Space %	High Volume Fragmentation Index	Largest Free Extent MB	Free Space MegaBytes
BENCHARC	Enabled	Pool	No	1	Enabled	99,3	0	8065	8066
CORZBSG	Enabled	Pool	No	96	Enabled	0,0	450	8105	197420
DWHZBSG	Enabled	Pool	No	16	Enabled	35,8	52	7841	108428
PRIVATE NON-SMS VOLUMES	Not_Connected	n/a	No	227	Enabled	0,0	586	8115	981597
STORAGE NON-SMS VOLUMES	Not_Connected	n/a	No	1	Enabled	24,3	205	630	1975
SYSSG	Enabled	Pool	No	10	Enabled	2,5	230	2920	20803
TRIADSG	Enabled	Pool	No	1	Enabled	18,8	124	1180	1534

Figure 3-8 The physical subtree of IBM OMAGEMON for Storage

The z/OS Storage Management Subsystem (SMS) is one of the key components in the host environment. With a good SMS, the system performs well. Otherwise, the system fails.

The OMEGAMON for Storage, as shown in Figure 3-8, provides a deep insight into the performance of this subsystem. These values are extremely focused on the requirements of the system programmers. Only a few applications that have high disk or tape activity are dedicated to a specific SMS group. In this case, the application context makes sense, but the Smart Bank showcase does not.

3.2 Physical views from IBM Tivoli Composite Application Manager

As described in “IBM Tivoli Composite Application Manager” on page 29, the Tivoli Composite Application Manager family is divided into the following multiple products:

- IBM Tivoli Composite Application Manager for Service-Oriented Architecture. This is described in 3.2.1, “IBM Tivoli Composite Application Manager for Service-Oriented Architecture” on page 132.
- IBM OMEGAMON for WebSphere Application Server on Distributed Systems and IBM OMEGAMON for WebSphere Application Server on z/OS. This is described in 3.2.2, “IBM OMEGAMON for WebSphere Application Server” on page 139.

- IBM OMEGAMON for WebSphere MQ. This is described in 3.2.3, “IBM OMEGAMON for WebSphere MQ” on page 149.

Note: In this project, OMEGAMON for WebSphere is used instead of IBM Tivoli Composite Application Manager for WebSphere, although IBM Tivoli Composite Application Manager for WebSphere is the designated successor of OMEGAMON for WebSphere. This is because the Tivoli Enterprise Monitoring Agent for IBM Tivoli Composite Application Manager/WebSphere Application Server was not available on z/OS at the time of writing this book. For further information, contact your nearest IBM sales representative.

For each of these products, the following questions are discussed:

- What can be expected from this agent?

  Each product is represented by a specific agent, a Tivoli Enterprise Monitoring Agent. The reported data is very specific to the application the agent is designed for.

- On which system must this specific agent type be installed?

  All agents will not be installed on every system of the infrastructure. Only those with a specific requirement will be injected by this agent.

- Which attributes does this agent support?

  Each agent type gathers its information into attribute groups. These groups are described, and a description of the problems that can be addressed by using this information is provided.

- Which are the key points to monitor from a system perspective?

  The different points of view within an IT organization may differ from team to team. The system group that is in charge of delivering a stable IT environment faces specific challenges.

- Which additional application requirements can be covered by the rules defined for this agent?

  Business unit managers who are in charge of a special part of the business, for example, the head of the online banking department, are less interested in single system components because their main focus is on the execution of their core business. They require designated information about the health of their supporting applications.

3.2.1 IBM Tivoli Composite Application Manager for Service-Oriented Architecture

IBM Tivoli Composite Application Manager for SOA comes with a predefined set of workspaces that is a part of the product. In this section, the workspaces are discussed in the way they appear in the physical navigation tree. All the window captures in this section are taken from the Smart Bank showcase development system.

For a detailed description of the agents’ workspace, consult this product’s user manual.

The Service Management Agent workspace

This workspace, shown in Figure 3-9, summarizes the actual monitoring settings for SOA services by the specific agent. The settings may be modified by the Take Action method (field A). This changes the amount of data gathered by the agent through the data collector inside the monitored application server.

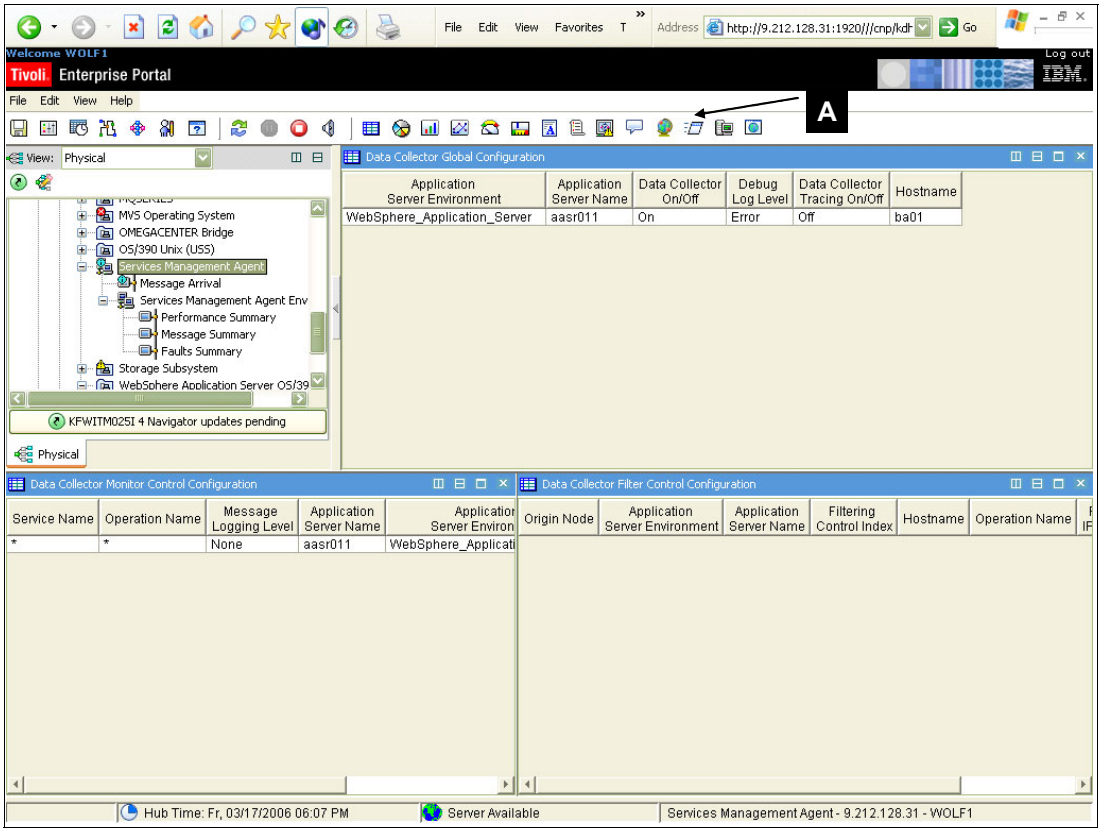


Figure 3-9 The Service Management Agent workspace

The Message Arrival workspace

This workspace, shown in Figure 3-10, offers a convenient summary of the number of messages that arrive in the data collector for each combination of service name, operation name, and remote IP address that are configured for monitoring. Because no modifications were made, all the messages are filtered by the product-predefined situation. The default shows the message arrival data for all the service names and operation names.

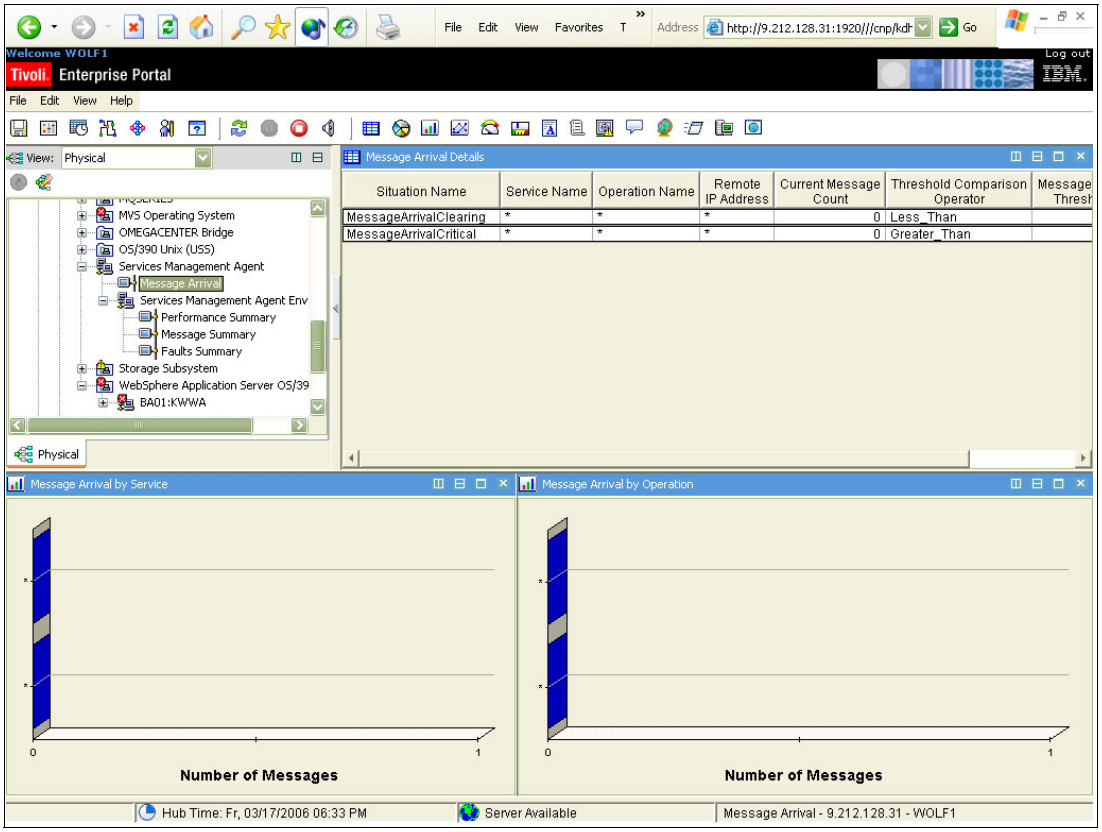


Figure 3-10 The Message Arrival workspace

The Service Management Environment workspace

This workspace, as shown in Figure 3-11, presents summarized data for all the monitored application servers on that system. Information from all the active data collectors are put together in order to provide a quick overview of SOA activity on this system. In the navigation tree, individual application servers are found.

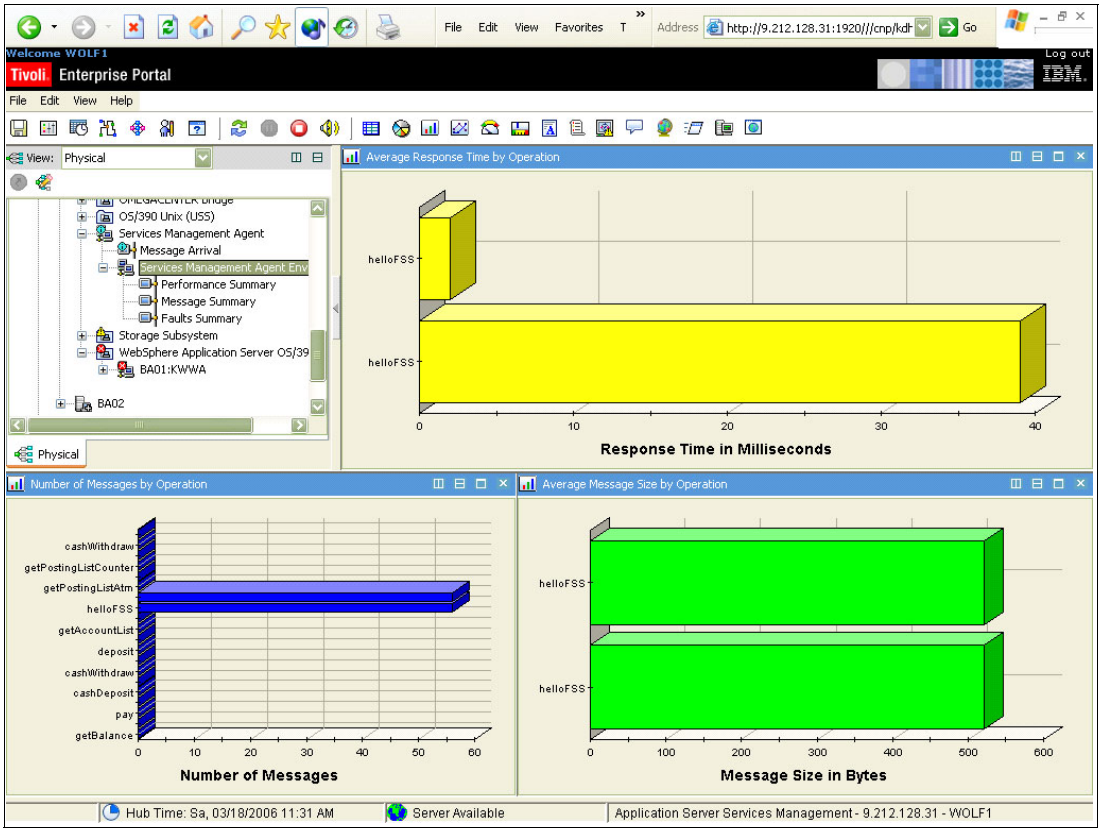


Figure 3-11 The Service Management Environment workspace

By default, the following key values are displayed:

- The average response taken time by the operation
- The number of messages by operation
- The fault summary by operation. This value is shown because there are multiple servers that are injected with the data collector on this single z/OS LPAR.

The Performance Summary workspace

This workspace is available for each application server injected with a data collector. As shown in Figure 3-12, it displays information about the response time of each called SOA service within the specific server.

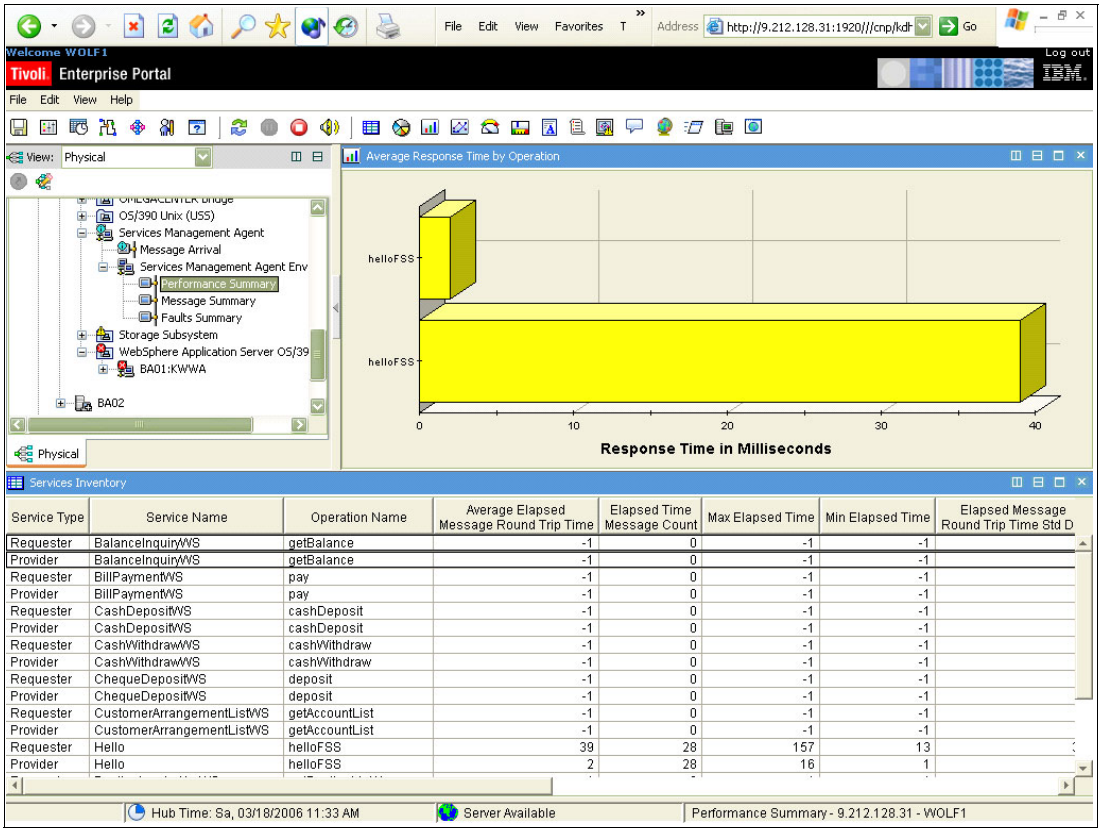

Welcome WOLF1
Tivoli Enterprise Portal
Log out
IBM
File Edit View Help
View: Physical
OS/390 Unix (USS)
Services Management Agent
Message Arrival
Services Management Agent Env
Performance Summary
Message Summary
Faults Summary
Storage Subsystem
WebSphere Application Server OS/39
BA01:KWWA
BA02
Physical
Average Response Time by Operation
helloFSS
helloFSS
0
10
20
30
40
Response Time in Milliseconds
Services Inventory
Service Type | Service Name | Operation Name | Average Elapsed Message Round Trip Time | Elapsed Time Message Count | Max Elapsed Time | Min Elapsed Time | Elapsed Message Round Trip Time Std D
Requester | BalanceInquiryWS | getBalance | -1 | 0 | -1 | -1
Provider | BalanceInquiryWS | getBalance | -1 | 0 | -1 | -1
Requester | BillPaymentWS | pay | -1 | 0 | -1 | -1
Provider | BillPaymentWS | pay | -1 | 0 | -1 | -1
Requester | CashDepositWS | cashDeposit | -1 | 0 | -1 | -1
Provider | CashDepositWS | cashDeposit | -1 | 0 | -1 | -1
Requester | CashWithdrawWS | cashWithdraw | -1 | 0 | -1 | -1
Provider | CashWithdrawWS | cashWithdraw | -1 | 0 | -1 | -1
Requester | ChequeDepositWS | deposit | -1 | 0 | -1 | -1
Provider | ChequeDepositWS | deposit | -1 | 0 | -1 | -1
Requester | CustomerArrangementListWS | getAccountList | -1 | 0 | -1 | -1
Provider | CustomerArrangementListWS | getAccountList | -1 | 0 | -1 | -1
Requester | Hello | helloFSS | 39 | 28 | 157 | 13
Provider | Hello | helloFSS | 2 | 28 | 16 | 1
Hub Time: Sa, 03/18/2006 11:33 AM
Server Available
Performance Summary - 9.212.128.31 - WOLF1

Figure 3-12 The Performance Summary workspace

By default, it is made up of two views:

- The average response time taken by the operation
- The services inventory

The Message Summary workspace

The Message Summary workspace, as shown in Figure 3-13, provides details about the number of messages and the average size of the messages (in bytes), by service name, operation name, and type (requester or provider). This workspace is available for each application server injected with the IBM Tivoli Composite Application Manager data collector for SOA.

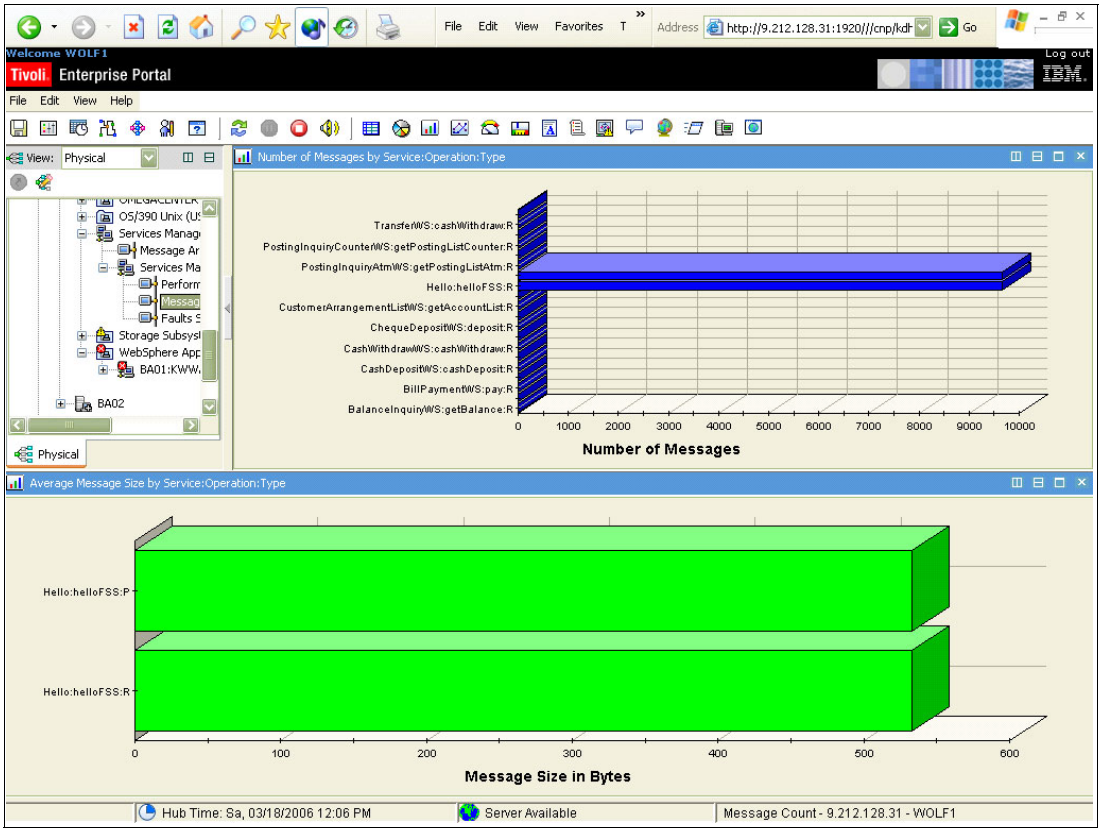


Figure 3-13 The Message Summary workspace

The Fault Summary workspace

The Fault Summary workspace provides a general summary of faults by operation. Figure 3-14 shows the workspace on the BA01 system.

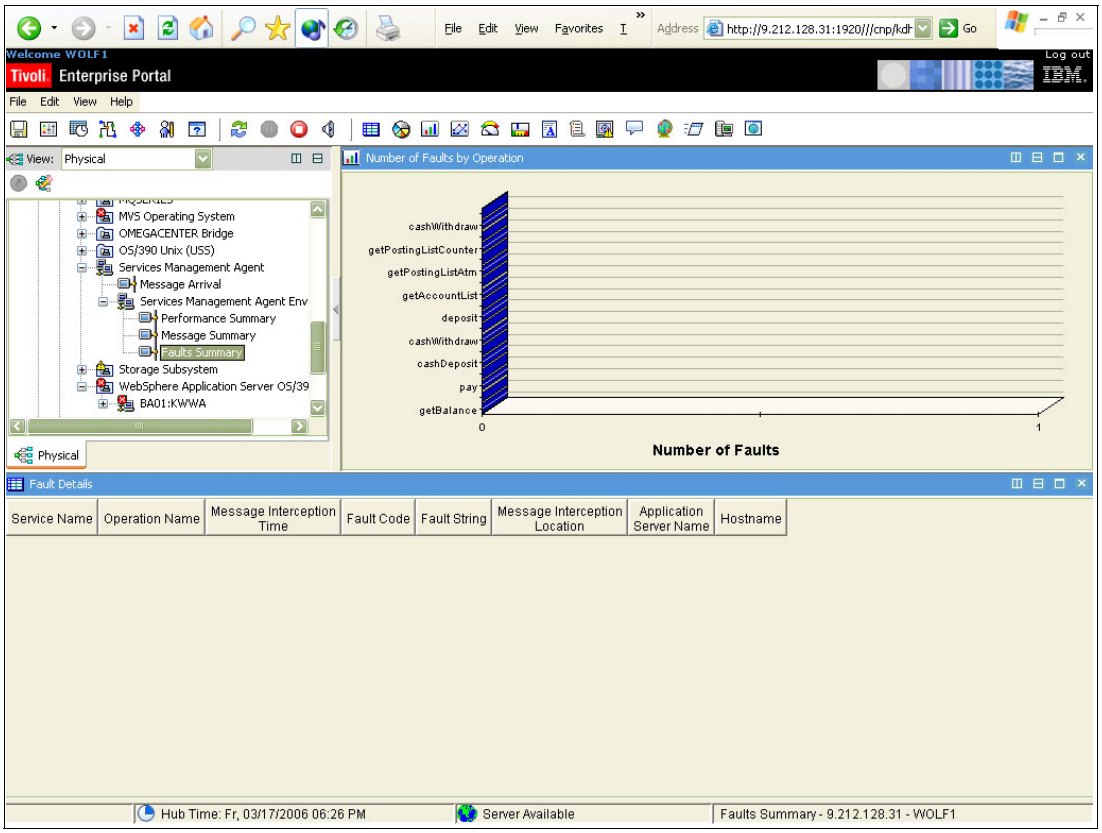


Figure 3-14 The Fault Summary workspace

By default there are two different views:

- The Fault Summary by Operation
- The Fault Details

This workspace is important for investigating errors and detecting upcoming issues.

IBM Tivoli Composite Application Manager for SOA: Summary

This product provides the management function managing the services in a service-oriented architecture running on application servers and WebSphere Service Integration Bus.

In the context of the Smart Bank showcase, it enables simple control of message traffic between Web services in the service-oriented architecture. The information flow between the services and the related response times and the failure occurrences become visible.

The agent is installed only on the z/OS where the SOA services are implemented. All the other systems are left untouched by this agent.

Because the SOA services are application-oriented, there are only a few specific system rules that are applicable to this kind of agent. The system operation can track only the accumulation of abnormal ends in SOA calls.

Monitoring values such as response times or number of messages is application-specific. These values must be applied in application-related rules. In most cases, application-specific rules are not attached to the physical tree. Such situations are typical when implementing the Logical view of an application within the monitoring environment.

3.2.2 IBM OMEGAMON for WebSphere Application Server

Because OMEGAMOM for WebSphere on distributed systems and OMEGAMON for WebSphere on z/OS have more or less the same functionality, the two products are described together. These products are replaced by IBM Tivoli Composite Application Manager for WebSphere Application Server described in 3.2.1, “IBM Tivoli Composite Application Manager for Service-Oriented Architecture” on page 132. Therefore, the differences between the functions are no longer visible under IBM Tivoli Monitoring V6 after migrating to IBM Tivoli Composite Application Manager for WebSphere. This includes the following features:

- Workload analysis
- Application trace

These functions belong to the deep-dive analysis category and are moved to the newly introduced IBM Tivoli Composite Application Manager managing server with its own user interface. This user interface addresses the requirements of application developers and deployers more than the operations and line business managers.

The OMEGAMON WebSphere Application Server agent enables the monitoring of the health and availability of WebSphere Application Server environments, as well as the performance of the deployed Java 2 Platform, Enterprise Edition (J2EE) applications inside the server.

The product is useful to execute the following tasks:

- To report current data and monitor performance for multiple platforms, operating systems, and applications
- To track and report the status of the WebSphere Application Server instances in your enterprise
- To centrally administer WebSphere Application Server in your enterprise, including starting and stopping servers and setting instrumentation levels in order to control performance data collection
- To collect and report traffic and resource use within each WebSphere Application Server, including the following:
  - Pool usage information
  - Java virtual machine (JVM) state
  - JVM memory
  - Garbage collection
  - Database connection
- To extensively monitor Enterprise JavaBeans (EJB), based on statistic measurement methods:
  - Report activity levels for each EJB container defined in each application server
  - Report activity levels for the enterprise bean object pools associated with enterprise beans
  - Collect and report statistics on EJB transactions
  - Report activity levels for individual enterprise beans, including method invocation data

- To get detailed statistical reports on Web application for each application server engine:
  - Using Servlets and Java Server Pages (JSP)
  - Hypertext Transfer Protocol (HTTP) sessions

OMEGAMON WebSphere Application Server users typically target the monitoring of the application server environment in the production environment. The monitoring expenses are low. Therefore, the costs involved in getting this information is low.

This section describes only the most important features of OMEGAMON for WebSphere Application Server for the Smart Bank showcase. Not all the workspaces are shown. The window captures are taken from the z/OS system BA01.

The OMEGAMON WebSphere top entry workspace

The workspace shown in Figure 3-15 is the default workspace.

- In area 1 shown in this figure, all the active WebSphere Application Server regions of the system in which the agent is running are displayed, including the WebSphere Application Server Deployment Manager and the Node Manager. This is the only area where these servers are visible.
- In area 2, the default IBM Tivoli Monitoring V6 enterprise console is displayed.

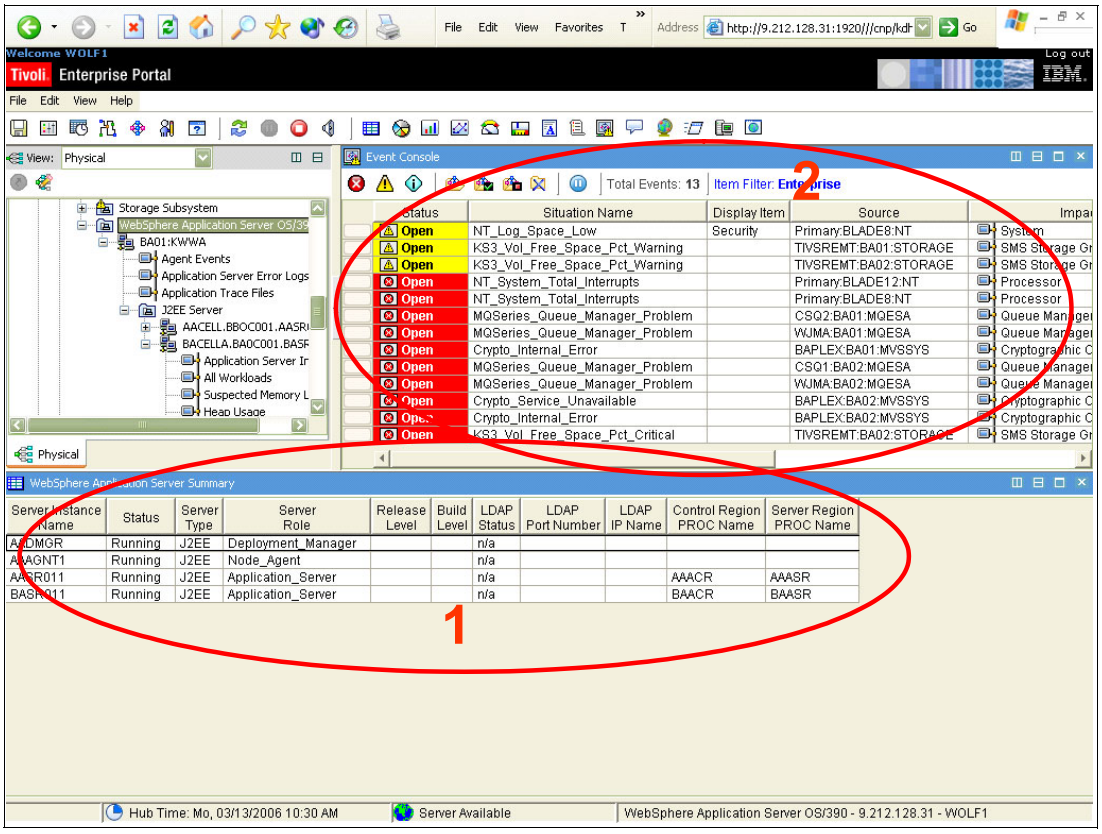


Figure 3-15 WebSphere Application Server z/OS default workspace

Figure 3-16 shows the customized workspace. It now includes the OMEGAMON WebSphere Application Server on z/OS Systems agent log and the WebSphere Application Server error log across all the monitored servers.

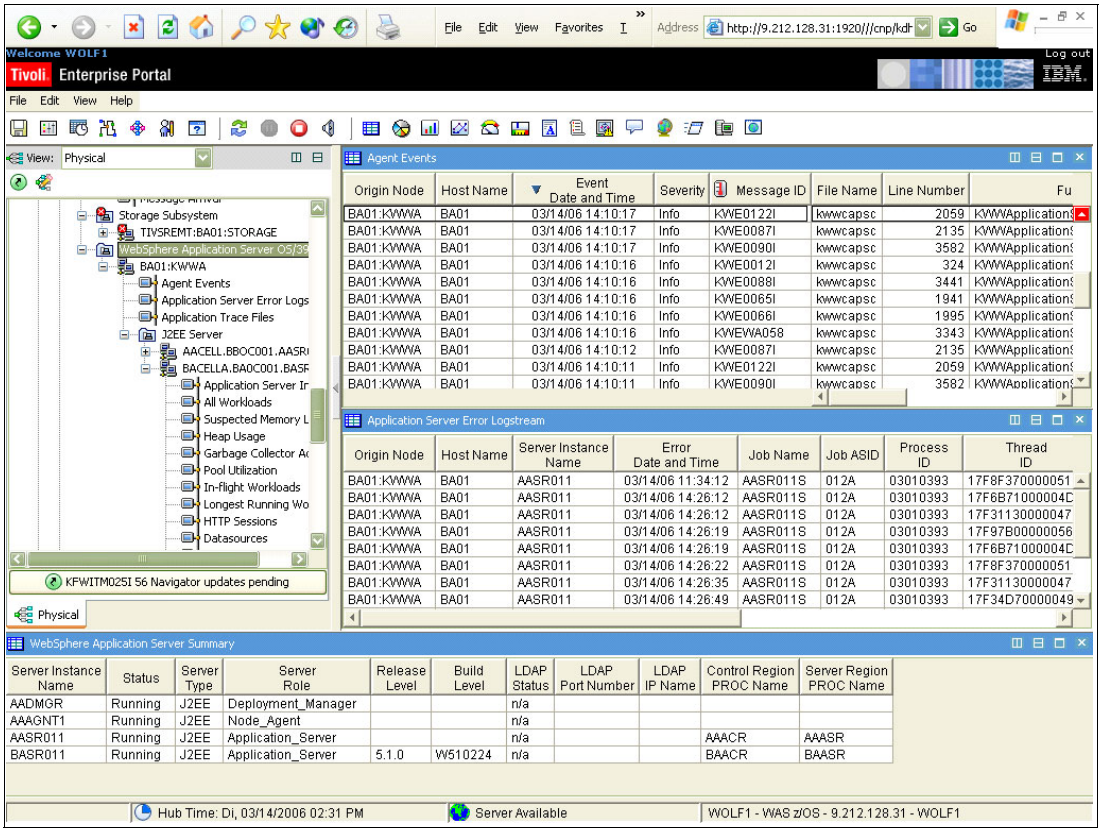


Figure 3-16 A customized OMEGAMON WebSphere Application Server z/OS Systems workspace

For a detailed description about customizing the workspace according to individual requirements, refer to Chapter 5 of the IBM Tivoli Monitoring V6 User Guide, which is available on the Web at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.itm.doc/itm610usersguide.pdf

For each server, a navigator item is available, with an identical subtree beneath it.

The Application Server Instance workspace

This workspace, as shown in Figure 3-17, provides an overview of the application server's health. It is a good place to associate rules that monitor the availability of the server instance. Associating (or disassociating) a situation means this situation will be attached to all the server subtrees to which this situation is distributed to.

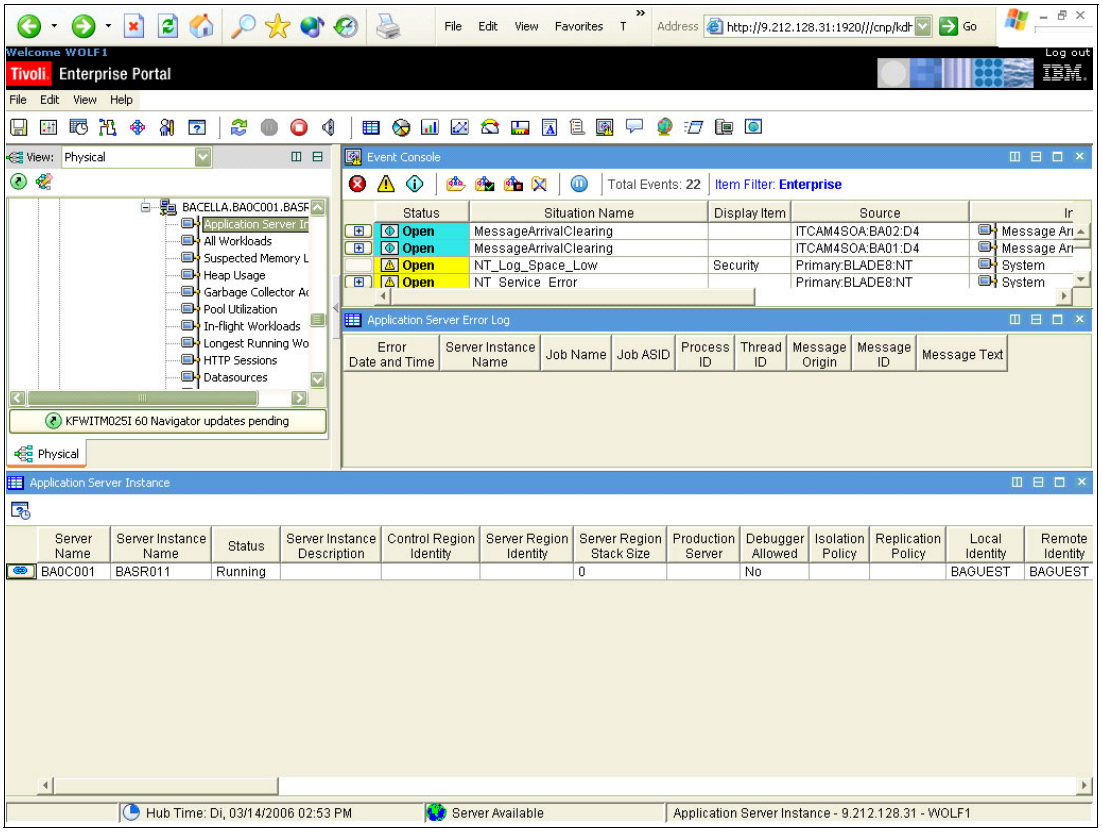


Figure 3-17 Application Server Instance workspace

The Garbage Collector Activity workspace

This workspace is important for system programmers. A high level of activity in the garbage collection workspace indicates a shortage of resources. The reason for this shortage must be investigated. On this workspace, you may find all the information required for getting closer to the root cause of the problem.

By placing a rule on the number of garbage collections rate (taken by the garbage collection by the minute), you can define a powerful alarm mechanism. Additionally, a look at the percentage of real time for garbage collection is useful.

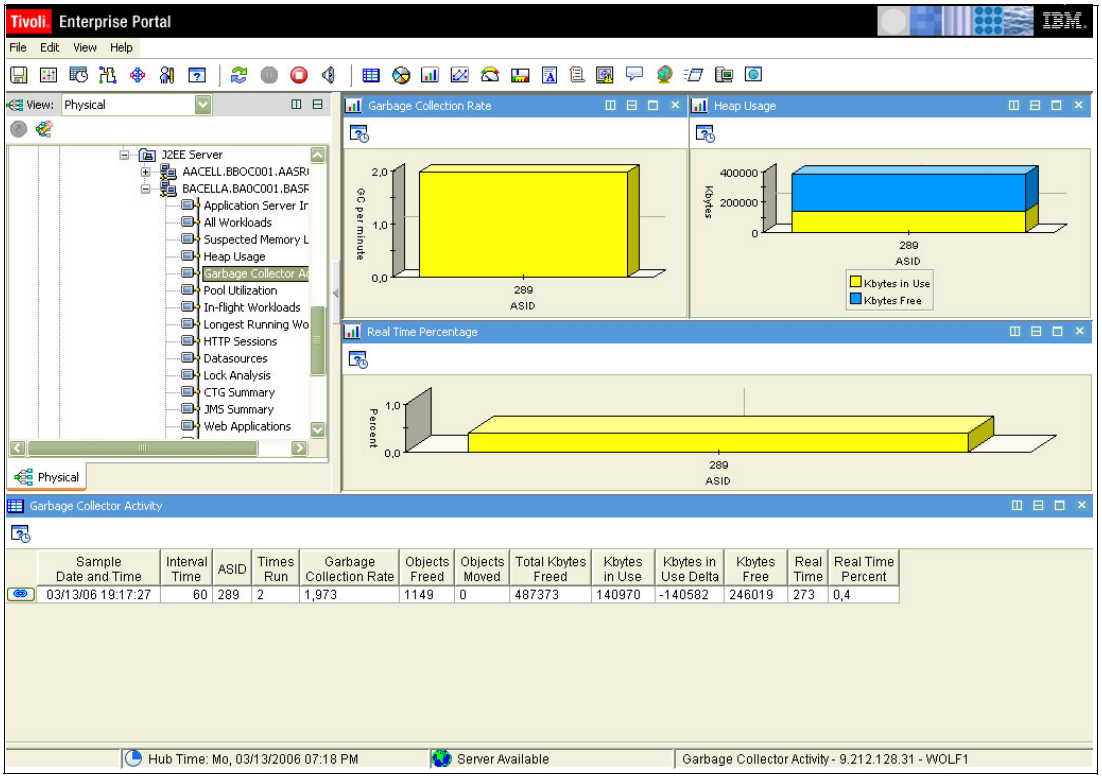


Figure 3-18 The Garbage Collector Activity workspace

A useful action based on this data is monitoring the number of HTTP sessions. Depending on the underlying system, this number may differ, but can be set by the system programmer's experience.

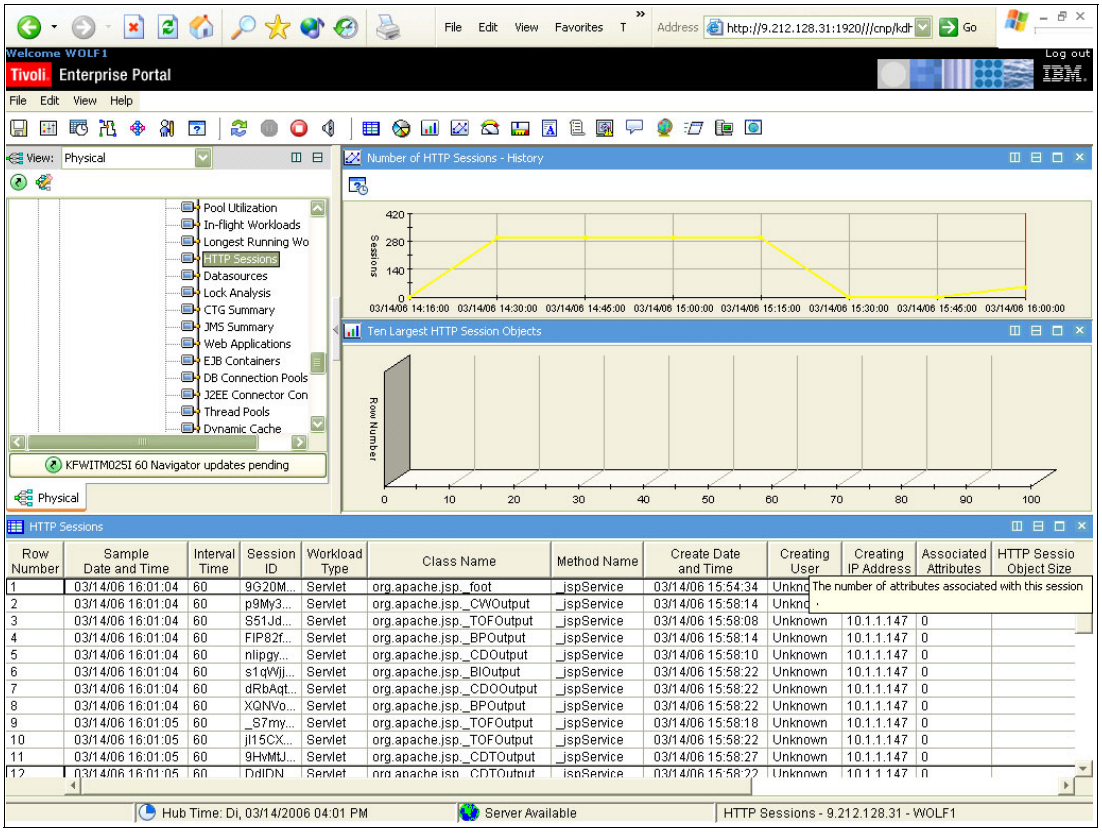
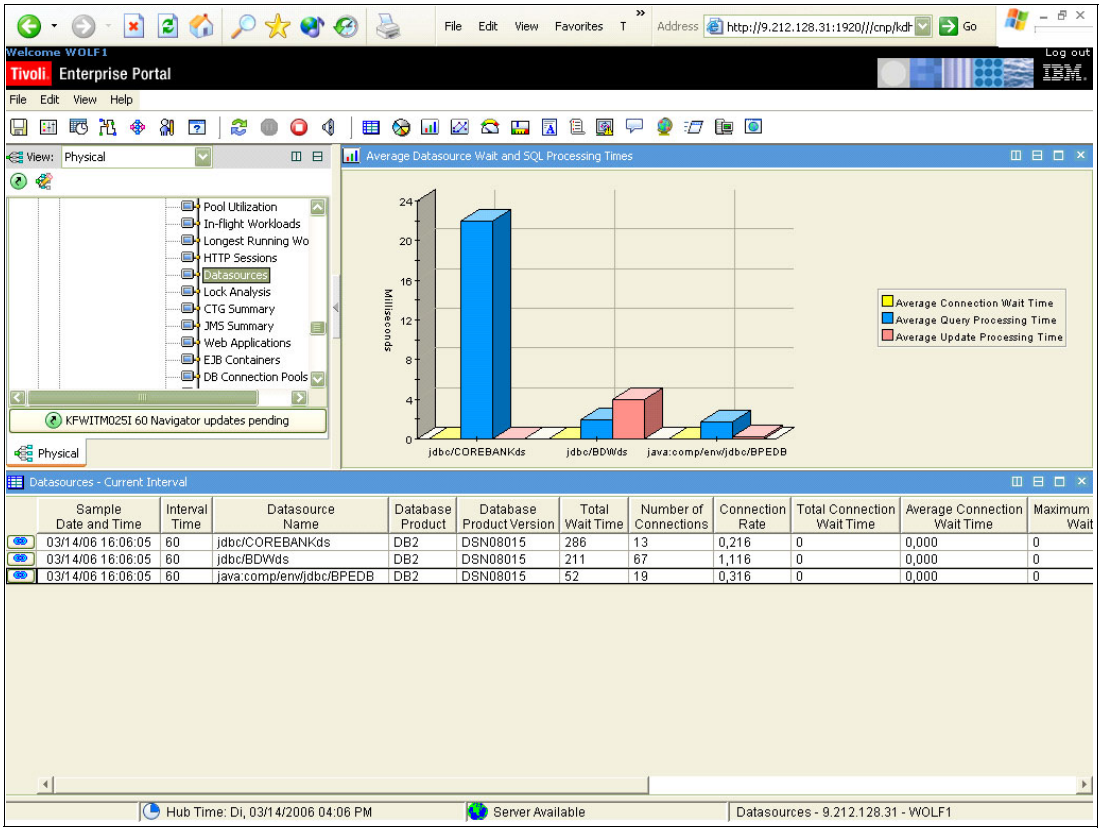

Welcome WOLF1
Tivoli Enterprise Portal
Log out
IBM
File Edit View Help
View: Physical
Pool Utilization
In-flight Workloads
Longest Running Wo
HTTP Sessions
Datasources
Lock Analysis
CTG Summary
JMS Summary
Web Applications
EJB Containers
DB Connection Pools
J2EE Connector Con
Thread Pools
Dynamic Cache
KFWITM025I 60 Navigator updates pending
Physical
Number of HTTP Sessions - History
Sessions
420
280
140
0
03/14/06 14:16:00 03/14/06 14:30:00 03/14/06 14:45:00 03/14/06 15:00:00 03/14/06 15:15:00 03/14/06 15:30:00 03/14/06 15:45:00 03/14/06 16:00:00
Ten Largest HTTP Session Objects
Row Number
0 10 20 30 40 50 60 70 80 90 100
HTTP Sessions
Row Number
Sample Date and Time
Interval Time
Session ID
Workload Type
Class Name
Method Name
Create Date and Time
Creating User
Creating IP Address
Associated Attributes
HTTP Sessio Object Size
The number of attributes associated with this session
1 03/14/06 16:01:04 60 9G20M... Servlet org.apache.jsp._foot _jspService 03/14/06 15:54:34 Unkno
2 03/14/06 16:01:04 60 p9My3... Servlet org.apache.jsp._CWOutput _jspService 03/14/06 15:58:14 Unkno
3 03/14/06 16:01:04 60 S51Jd... Servlet org.apache.jsp._TOFOutput _jspService 03/14/06 15:58:08 Unknown 10.1.1.147 0
4 03/14/06 16:01:04 60 FIP82f... Servlet org.apache.jsp._BPOutput _jspService 03/14/06 15:58:14 Unknown 10.1.1.147 0
5 03/14/06 16:01:04 60 nlipgy... Servlet org.apache.jsp._CDOutput _jspService 03/14/06 15:58:10 Unknown 10.1.1.147 0
6 03/14/06 16:01:04 60 s1qWjj... Servlet org.apache.jsp._BIOutput _jspService 03/14/06 15:58:22 Unknown 10.1.1.147 0
7 03/14/06 16:01:04 60 dRbAqt... Servlet org.apache.jsp._CDOOutput _jspService 03/14/06 15:58:22 Unknown 10.1.1.147 0
8 03/14/06 16:01:04 60 XQNVo... Servlet org.apache.jsp._BPOutput _jspService 03/14/06 15:58:22 Unknown 10.1.1.147 0
9 03/14/06 16:01:05 60 _S7my... Servlet org.apache.jsp._TOFOutput _jspService 03/14/06 15:58:18 Unknown 10.1.1.147 0
10 03/14/06 16:01:05 60 jl15CX... Servlet org.apache.jsp._TOFOutput _jspService 03/14/06 15:58:22 Unknown 10.1.1.147 0
11 03/14/06 16:01:05 60 9HvMtJ... Servlet org.apache.jsp._CDTOutput _jspService 03/14/06 15:58:27 Unknown 10.1.1.147 0
12 03/14/06 16:01:05 60 DdIDN... Servlet org.apache.jsp._CDTOutput _jspService 03/14/06 15:58:22 Unknown 10.1.1.147 0
Hub Time: Di, 03/14/2006 04:01 PM
Server Available
HTTP Sessions - 9.212.128.31 - WOLF1

Figure 3-19 The HTTP Sessions workspace

The Datasource workspace

This workspace, shown in Figure 3-20, provides a brief overview of defined data source usage in the WebSphere Application Server.

One of the useful situations involves testing the existence of all the required data sources. Another important rule is to look at the Average Connection Wait Time, which indicates that the request going to this data source is in the queue before it is executed.



Sample Date and Time	Interval Time	Datasource Name	Database Product	Database Product Version	Total Wait Time	Number of Connections	Connection Rate	Total Connection Wait Time	Average Connection Wait Time	Maximum Wait
03/14/06 16:06:05	60	jdbc/COREBANKds	DB2	DSN08015	286	13	0,216	0	0,000	0
03/14/06 16:06:05	60	jdbc/BDWds	DB2	DSN08015	211	67	1,116	0	0,000	0
03/14/06 16:06:05	60	java:comp/env/jdbc/BPEDB	DB2	DSN08015	52	19	0,316	0	0,000	0

Figure 3-20 The data source workspace

The Web Applications workspace

This workspace, shown in Figure 3-21, provides an overview of all the servlets and JavaServer Pages executed in the selected Web container. This workspace is application-focused. The response time given is specific to each application. This means that no system-wide rule can be applied, except for a general rule such as no transaction must, on an average, last for more than xx milliseconds.

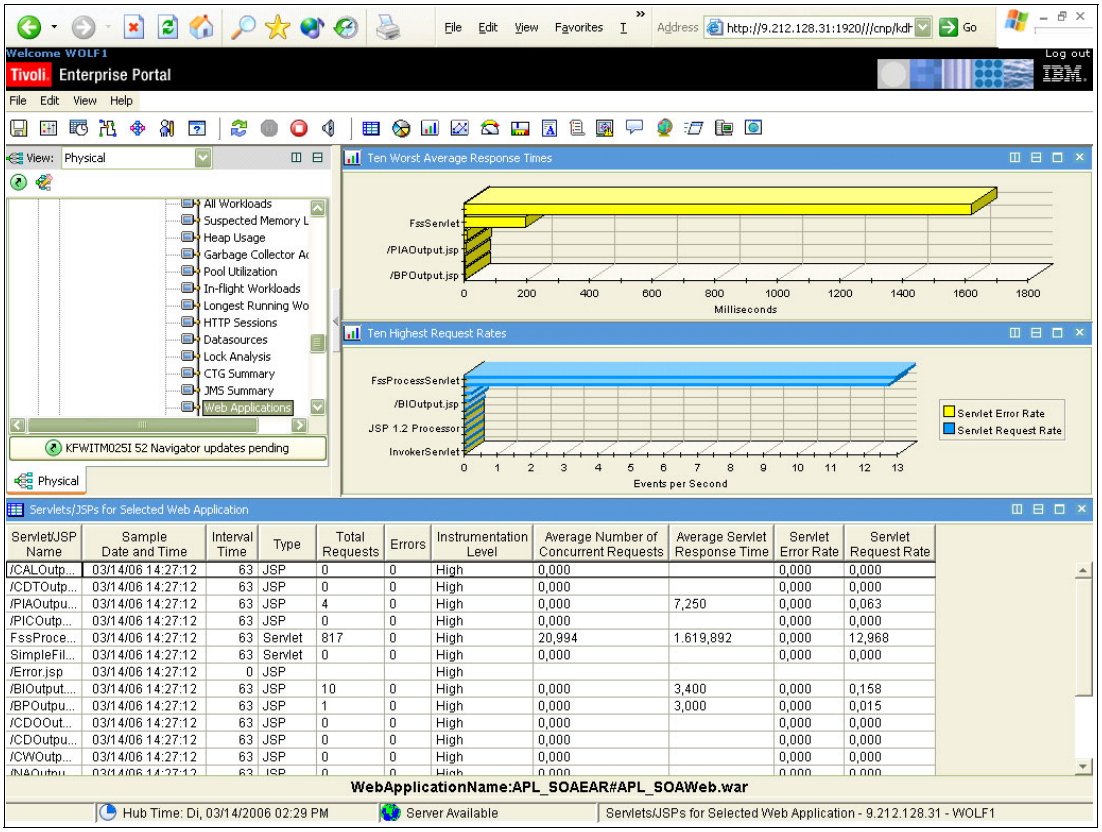

Welcome WOLF1
Tivoli Enterprise Portal
Log out
File Edit View Help
View: Physical
All Workloads
Suspected Memory L
Heap Usage
Garbage Collector Ac
Pool Utilization
In-flight Workloads
Longest Running Wo
HTTP Sessions
Datasources
Lock Analysis
CTG Summary
JMS Summary
Web Applications
KFWITM025I 52 Navigator updates pending
Physical
Ten Worst Average Response Times
FssServlet
/PIAOutput.jsp
/BPOutput.jsp
0 200 400 600 800 1000 1200 1400 1600 1800
Milliseconds
Ten Highest Request Rates
FssProcessServlet
/BIOutput.jsp
JSP 1.2 Processor
InvokerServlet
0 1 2 3 4 5 6 7 8 9 10 11 12 13
Events per Second
Servlet Error Rate
Servlet Request Rate
Servlets/JSPs for Selected Web Application
Servlet/JSP Name | Sample Date and Time | Interval Time | Type | Total Requests | Errors | Instrumentation Level | Average Number of Concurrent Requests | Average Servlet Response Time | Servlet Error Rate | Servlet Request Rate
/CALOutp... | 03/14/06 14:27:12 | 63 | JSP | 0 | 0 | High | 0,000 | | 0,000 | 0,000
/CDTOutp... | 03/14/06 14:27:12 | 63 | JSP | 0 | 0 | High | 0,000 | | 0,000 | 0,000
/PIAOutpu... | 03/14/06 14:27:12 | 63 | JSP | 4 | 0 | High | 0,000 | 7,250 | 0,000 | 0,063
/PICOutp... | 03/14/06 14:27:12 | 63 | JSP | 0 | 0 | High | 0,000 | | 0,000 | 0,000
FssProce... | 03/14/06 14:27:12 | 63 | Servlet | 817 | 0 | High | 20,994 | 1.619,892 | 0,000 | 12,968
SimpleFil... | 03/14/06 14:27:12 | 63 | Servlet | 0 | 0 | High | 0,000 | | 0,000 | 0,000
/Error.jsp | 03/14/06 14:27:12 | 0 | JSP | | | High | | | |
/BIOutput... | 03/14/06 14:27:12 | 63 | JSP | 10 | 0 | High | 0,000 | 3,400 | 0,000 | 0,158
/BPOutpu... | 03/14/06 14:27:12 | 63 | JSP | 1 | 0 | High | 0,000 | 3,000 | 0,000 | 0,015
/CDOOut... | 03/14/06 14:27:12 | 63 | JSP | 0 | 0 | High | 0,000 | | 0,000 | 0,000
/CDOutpu... | 03/14/06 14:27:12 | 63 | JSP | 0 | 0 | High | 0,000 | | 0,000 | 0,000
/CWOutp... | 03/14/06 14:27:12 | 63 | JSP | 0 | 0 | High | 0,000 | | 0,000 | 0,000
WebApplicationName:APL_SOAEAR#APL_SOAWeb.war
Hub Time: Di, 03/14/2006 02:29 PM
Server Available
Servlets/JSPs for Selected Web Application - 9.212.128.31 - WOLF1

Figure 3-21 The Web Applications workspace

On the other hand, the servlet error rate is also useful for system programmers. Faulty applications may result in poor overall system performance, higher rollback activity, and so on.

Different rules may be applied, depending on the audience:

- Rules for system programmers:
  - Servlet error rate, which indicates a faulty application or an absence of a system resource
  - Total request rate, which indicates no activity for this application or vice versa, that is, an overload of this application
- Rules for business owners:
  - Web application-specific response time limits
  - Application-specific servlet request rate

These rules are implemented in our Smart Bank showcase. Make a note of the fact that each client may have individual requirements.

The Enterprise JavaBeans Container workspace

The EJB Container workspace, as shown in Figure 3-22, provides information about the EJBs that are executed. It is application-specific.

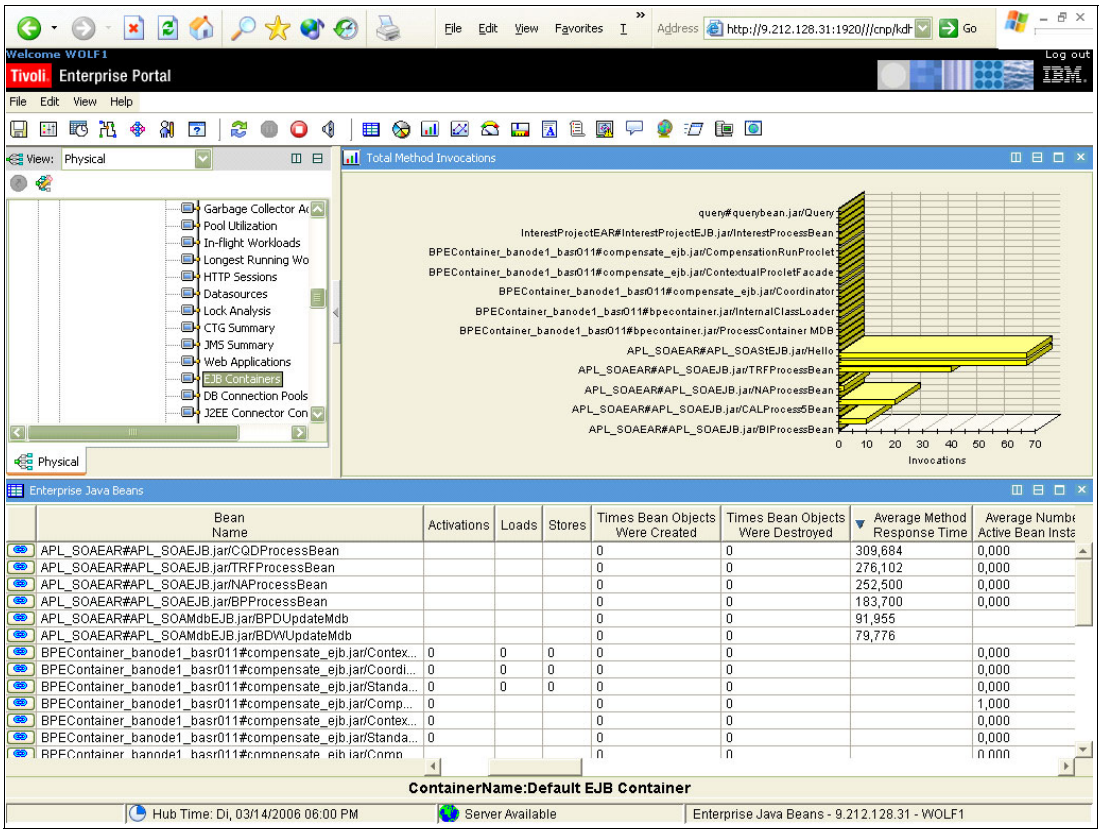

Welcome WOLF1
Tivoli Enterprise Portal
File Edit View Help
View: Physical
Garbage Collector Ac
Pool Utilization
In-flight Workloads
Longest Running Wo
HTTP Sessions
Datasources
Lock Analysis
CTG Summary
JMS Summary
Web Applications
EJB Containers
DB Connection Pools
J2EE Connector Con
Physical
Total Method Invocations
query#querybean.jar/Query
InterestProjectEAR#InterestProjectEJB.jar/InterestProcessBean
BPEContainer_banode1_basr011#compensate_ejb.jar/CompensationRunProclet
BPEContainer_banode1_basr011#compensate_ejb.jar/ContextualProcletFacade
BPEContainer_banode1_basr011#compensate_ejb.jar/Coordinator
BPEContainer_banode1_basr011#bpecontainer.jar/InternalClassLoader
BPEContainer_banode1_basr011#bpecontainer.jar/ProcessContainer MDB
APL_SOAEAR#APL_SOAStEJB.jar/Hello
APL_SOAEAR#APL_SOAEJB.jar/TRFProcessBean
APL_SOAEAR#APL_SOAEJB.jar/NAProcessBean
APL_SOAEAR#APL_SOAEJB.jar/CALProcess5Bean
APL_SOAEAR#APL_SOAEJB.jar/BIProcessBean
0 10 20 30 40 50 60 70
Invocations
Enterprise Java Beans
Bean Name | Activations | Loads | Stores | Times Bean Objects Were Created | Times Bean Objects Were Destroyed | Average Method Response Time | Average Numbe Active Bean Insta
APL_SOAEAR#APL_SOAEJB.jar/CQDProcessBean | 0 | 309,684 | 0,000
APL_SOAEAR#APL_SOAEJB.jar/TRFProcessBean | 0 | 276,102 | 0,000
APL_SOAEAR#APL_SOAEJB.jar/NAProcessBean | 0 | 252,500 | 0,000
APL_SOAEAR#APL_SOAEJB.jar/BPProcessBean | 0 | 183,700 | 0,000
APL_SOAEAR#APL_SOAMdbEJB.jar/BPDUpdateMdb | 0 | 91,955
APL_SOAEAR#APL_SOAMdbEJB.jar/BDWUpdateMdb | 0 | 79,776
BPEContainer_banode1_basr011#compensate_ejb.jar/Contex... | 0 | 0 | 0 | 0 | 0 | 0,000
BPEContainer_banode1_basr011#compensate_ejb.jar/Coordi... | 0 | 0 | 0 | 0 | 0 | 0,000
BPEContainer_banode1_basr011#compensate_ejb.jar/Standa... | 0 | 0 | 0 | 0 | 0 | 0,000
BPEContainer_banode1_basr011#compensate_ejb.jar/Comp... | 0 | 0 | 0 | 0 | 1,000
BPEContainer_banode1_basr011#compensate_ejb.jar/Contex... | 0 | 0 | 0 | 0 | 0,000
BPEContainer_banode1_basr011#compensate_ejb.jar/Standa... | 0 | 0 | 0 | 0 | 0,000
ContainerName:Default EJB Container
Hub Time: Di, 03/14/2006 06:00 PM
Server Available
Enterprise Java Beans - 9.212.128.31 - WOLF1

Figure 3-22 The EJB Container workspace

Web applications using EJBs are measured from the single methods that are invoked by the Web applications (or other EJBs). An EJB by itself does not have a response time.

Set the following rules:

- For system programmers:
  - None at this time
- For business managers:
  - EJB-specific response time measurements
  - EJB-specific invocation rate

Most of the time, these situations are client-specific.

The DB Connection Pool workspace

This workspace, as shown in Figure 3-23, provides details about the information provided by the Data source workspace. In this workspace, only the data sources to the database are displayed, but with more specific information.

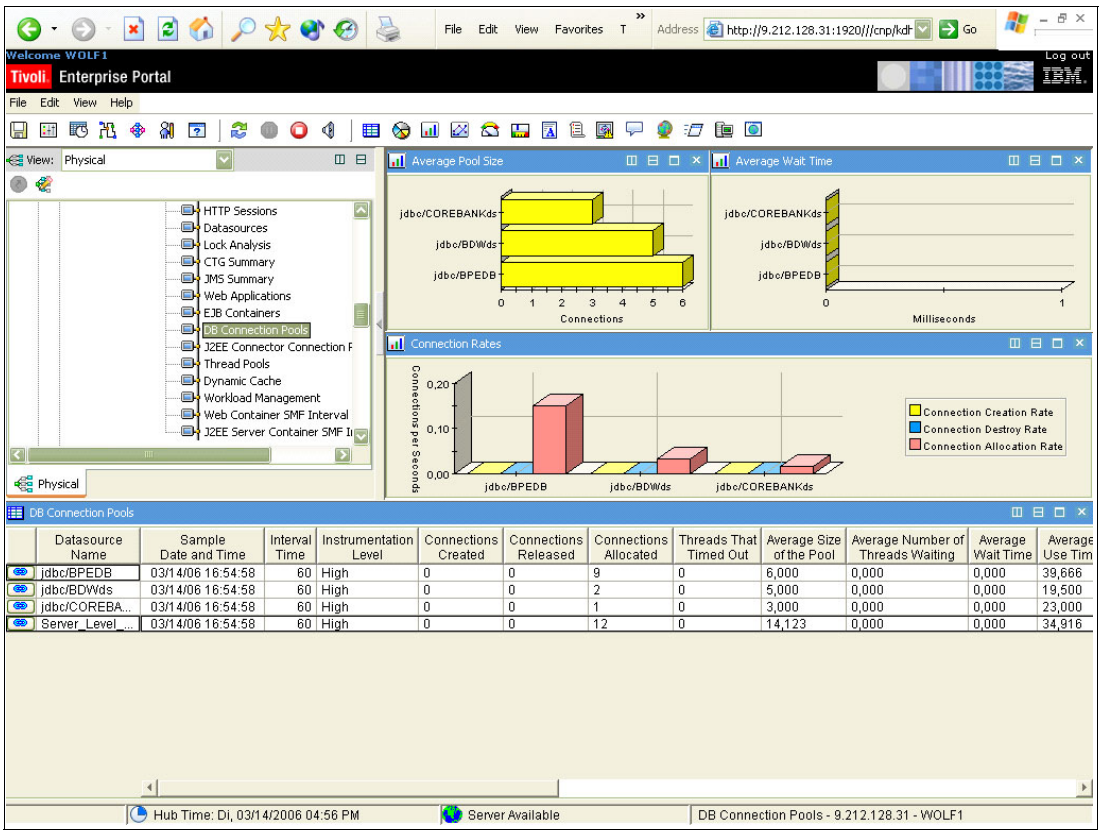


Figure 3-23 The DB Connection Pool workspace

Set the following rules:

- For system programmers:
  - Percentage of the pool connections used
  - The average wait time
- For business managers:

  If database connections that are specific to a single application are defined, the same situations used in the case of system programmers may be attached to the Logical view.

OMEGAMON for WebSphere Application Server: Summary

This product provides powerful functions for the operations community as well as for application managers to keep track of the performance of the components they are in charge of.

The agents are installed on z/OS and Linux where the WebSphere Application Server is running on. Because both the platforms are measured almost identically and the output is almost the same, this product provides real value to the Smart Bank showcase and their requirements for platform spanning, cross system monitoring, and subsystem monitoring.

The monitor for WebSphere provides information about the subsystem and the application running inside. It provides information about the incorporated service providers and the interfaces, such as J2EE Connector architecture (JCA), Java Database Connectivity (JDBC™), Java Message Service (JMS), and CICS Transaction Gateway (CTG).

In the new IBM Tivoli Composite Application Manager for WebSphere, a deep dive analysis mechanism has been added.

3.2.3 IBM OMEGAMON for WebSphere MQ

The OMEGAMON for WebSphere MQ agent enables in-depth monitoring of the IBM MQSeries® subsystem, its implementing objects, and the messages currently contained in the MQ system.

OMEGAMON for WebSphere MQ is more or less the same across all the supported platforms. It always displays the same navigation tree, independent of the platform it is running on. Some workspaces may stay empty because data is not available for that platform. All the rules defined for this agent type can be spread across all the underlying platforms.

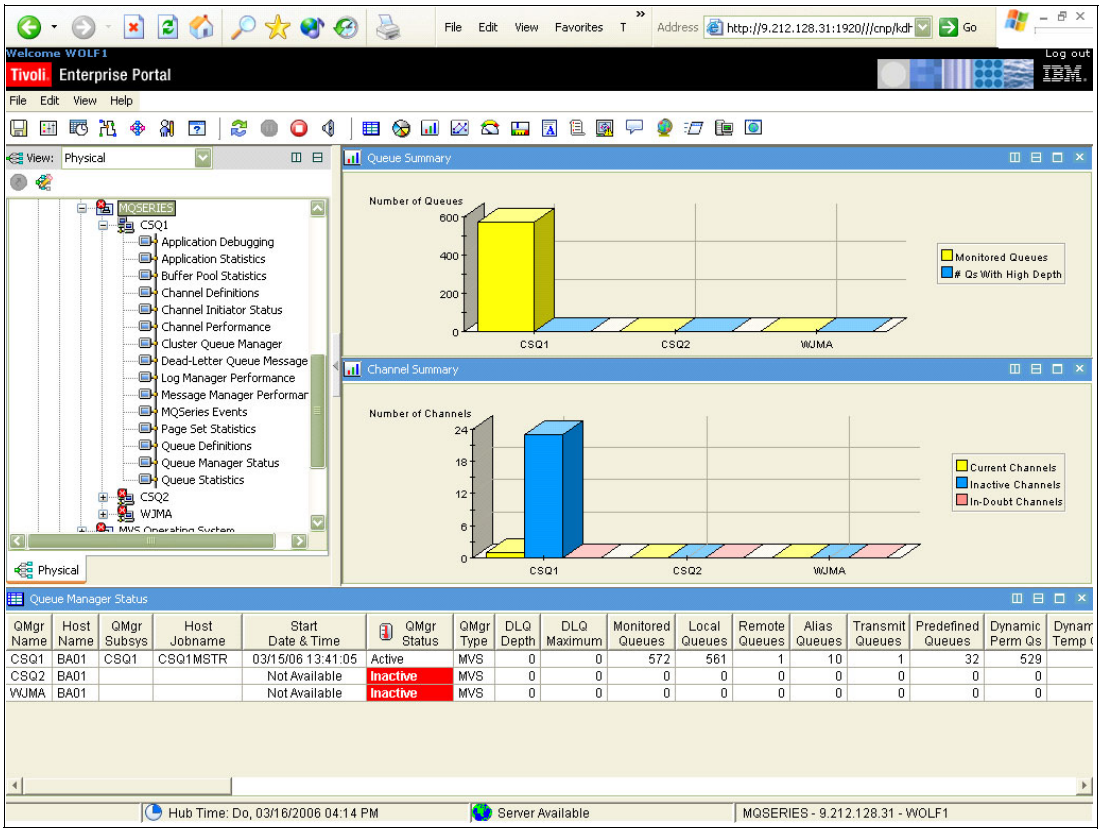

Welcome WOLF1
Tivoli Enterprise Portal
File Edit View Help
View: Physical
MQSERIES
CSQ1
Application Debugging
Application Statistics
Buffer Pool Statistics
Channel Definitions
Channel Initiator Status
Channel Performance
Cluster Queue Manager
Dead-Letter Queue Message
Log Manager Performance
Message Manager Performar
MQSeries Events
Page Set Statistics
Queue Definitions
Queue Manager Status
Queue Statistics
CSQ2
WJMA
Physical
Queue Summary
Number of Queues
600
400
200
0
CSQ1
CSQ2
WJMA
Monitored Queues
Qs With High Depth
Channel Summary
Number of Channels
24
18
12
6
0
CSQ1
CSQ2
WJMA
Current Channels
Inactive Channels
In-Doubt Channels
Queue Manager Status
QMgr Name | Host Name | QMgr Subsys | Host Jobname | Start Date & Time | QMgr Status | QMgr Type | DLQ Depth | DLQ Maximum | Monitored Queues | Local Queues | Remote Queues | Alias Queues | Transmit Queues | Predefined Queues | Dynamic Perm Qs
CSQ1 | BA01 | CSQ1 | CSQ1MSTR | 03/15/06 13:41:05 | Active | MVS | 0 | 0 | 572 | 561 | 1 | 10 | 1 | 32 | 529
CSQ2 | BA01 | | | Not Available | Inactive | MVS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
WJMA | BA01 | | | Not Available | Inactive | MVS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0
Hub Time: Do, 03/16/2006 04:14 PM
Server Available
MQSERIES - 9.212.128.31 - WOLF1

Figure 3-24 The physical tree of OMEGAMON for WebSphere MQ monitoring

The following workspaces exist for z/OS only:

- Application statistics
- Application debugging
- Buffer pool statistics
- Channel initiator status
- Message manager performance
- Page set statistics

The Error Log Workspace is *not* available on z/OS systems. That is why it is not displayed in the window shown in Figure 3-20 on page 145.

The OMEGAMON for WebSphere MQ product is useful for executing the following tasks:

- Collecting and analyzing MQSeries-specific data for all your remote and local WebSphere MQ queue managers from a single vantage point
- Providing many useful workspaces that you can use to track trends and understand system problems
- Providing the ability to view information about each WebSphere MQ system you are monitoring. This information is useful for:
  - Monitoring the performance of each WebSphere MQ-managed system and helping identify system bottlenecks and evaluating tuning decisions
  - Selecting the most effective threshold values
  - Reviewing the status information when a change in the state of a given resource occurs

All the workspaces in OMEGAMON for WebSphere MQ are important to set up successful enterprise-wide monitoring. WebSphere MQ is spread across the entire company to deliver a common layer for message exchange. The monitoring is also spread over these systems. The performance and efficiency of this system is a key point for a performing application environment.

"The Channel Performance workspace" on page 151 and "The Queue Statistics workspace" on page 152 demonstrate how varied monitoring requirements between system programmers can be, and how they are supported by IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6.

The Channel Performance workspace

Channels are related to queue managers and their interconnection. Except for client channels, which are often application-specific, most of the rules applied for channel performance are system focused. A Channel Performance workspace is shown in Figure 3-25.

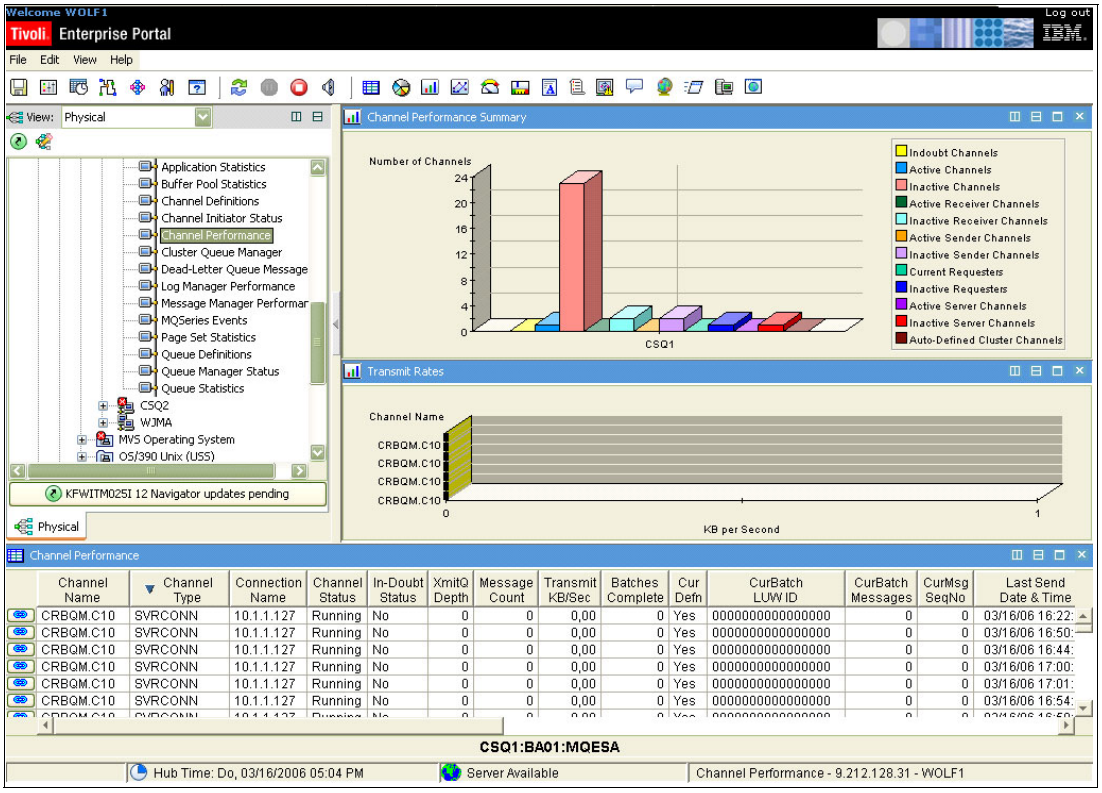

Welcome WOLF1
Tivoli Enterprise Portal
Log out
File Edit View Help
View: Physical
Application Statistics
Buffer Pool Statistics
Channel Definitions
Channel Initiator Status
Channel Performance
Cluster Queue Manager
Dead-Letter Queue Message
Log Manager Performance
Message Manager Performar
MQSeries Events
Page Set Statistics
Queue Definitions
Queue Manager Status
Queue Statistics
CSQ2
WJMA
MVS Operating System
OS/390 Unix (USS)
KFWITM025I 12 Navigator updates pending
Physical
Channel Performance Summary
Number of Channels
CSQ1
Indoubt Channels
Active Channels
Inactive Channels
Active Receiver Channels
Inactive Receiver Channels
Active Sender Channels
Inactive Sender Channels
Current Requesters
Inactive Requesters
Active Server Channels
Inactive Server Channels
Auto-Defined Cluster Channels
Transmit Rates
Channel Name
CRBQM.C10
KB per Second
Channel Performance
Channel Name | Channel Type | Connection Name | Channel Status | In-Doubt Status | XmitQ Depth | Message Count | Transmit KB/Sec | Batches Complete | Cur Defn | CurBatch LUW ID | CurBatch Messages | CurMsg SeqNo | Last Send Date & Time
CRBQM.C10 | SVRCONN | 10.1.1.127 | Running | No | 0 | 0 | 0,00 | 0 | Yes | 0000000000000000 | 0 | 0 | 03/16/06 16:22:
CRBQM.C10 | SVRCONN | 10.1.1.127 | Running | No | 0 | 0 | 0,00 | 0 | Yes | 0000000000000000 | 0 | 0 | 03/16/06 16:50:
CRBQM.C10 | SVRCONN | 10.1.1.127 | Running | No | 0 | 0 | 0,00 | 0 | Yes | 0000000000000000 | 0 | 0 | 03/16/06 16:44:
CRBQM.C10 | SVRCONN | 10.1.1.127 | Running | No | 0 | 0 | 0,00 | 0 | Yes | 0000000000000000 | 0 | 0 | 03/16/06 17:00:
CRBQM.C10 | SVRCONN | 10.1.1.127 | Running | No | 0 | 0 | 0,00 | 0 | Yes | 0000000000000000 | 0 | 0 | 03/16/06 17:01:
CRBQM.C10 | SVRCONN | 10.1.1.127 | Running | No | 0 | 0 | 0,00 | 0 | Yes | 0000000000000000 | 0 | 0 | 03/16/06 16:54:
CSQ1:BA01:MQESA
Hub Time: Do, 03/16/2006 05:04 PM
Server Available
Channel Performance - 9.212.128.31 - WOLF1

Figure 3-25 The Channel Performance workspace

The system programmer may have to identify the upcoming threads earlier on in order to have enough time to fix the issues before the applications or the entire system collapses.

We recommend that you perform the following checks:

- A channel is not up and running when there are messages on the transmission queue

  Only an empty transmission queue is a good queue. Ask yourself why the channel is not starting automatically.

- A channel gets into the “in doubt status” for two or more test intervals

  This indicates, for example, that the channel connection is either unstable due to communication protocol problems or WebSphere MQ cannot recover the in doubt state by itself.

- Check the channel status

  These channels cannot start up automatically. They must be activated by commands.

These are only a few suggestions. Applications are not aware of the MQ infrastructure they use.

The Queue Statistics workspace

This workspace, shown in Figure 3-26, provides detailed information about each queue and its usage. The system aspects are linked to the numbers provided in relative usage amount, but the application community is more interested in detailed, single numbers for specific queues. Both the user groups are supported here.

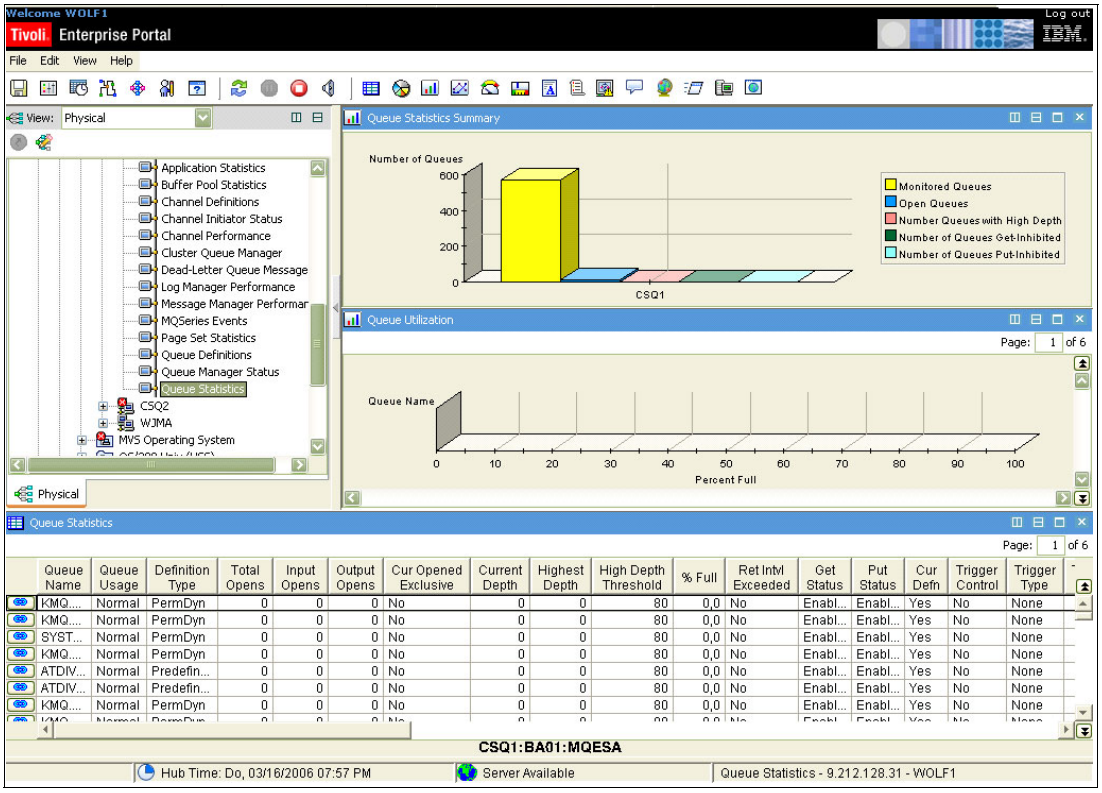

Welcome WOLF1
Tivoli Enterprise Portal
Log out
File Edit View Help
View: Physical
Application Statistics
Buffer Pool Statistics
Channel Definitions
Channel Initiator Status
Channel Performance
Cluster Queue Manager
Dead-Letter Queue Message
Log Manager Performance
Message Manager Performar
MQSeries Events
Page Set Statistics
Queue Definitions
Queue Manager Status
Queue Statistics
CSQ2
WJMA
MVS Operating System
Physical
Queue Statistics Summary
Number of Queues
600
400
200
0
CSQ1
Monitored Queues
Open Queues
Number Queues with High Depth
Number of Queues Get-Inhibited
Number of Queues Put-Inhibited
Queue Utilization
Page: 1 of 6
Queue Name
0 10 20 30 40 50 60 70 80 90 100
Percent Full
Queue Statistics
Page: 1 of 6
Queue Name | Queue Usage | Definition Type | Total Opens | Input Opens | Output Opens | Cur Opened Exclusive | Current Depth | Highest Depth | High Depth Threshold | % Full | Ret Intvl Exceeded | Get Status | Put Status | Cur Defn | Trigger Control | Trigger Type
KMQ... | Normal | PermDyn | 0 | 0 | 0 | No | 0 | 0 | 80 | 0,0 | No | Enabl... | Enabl... | Yes | No | None
KMQ... | Normal | PermDyn | 0 | 0 | 0 | No | 0 | 0 | 80 | 0,0 | No | Enabl... | Enabl... | Yes | No | None
SYST... | Normal | PermDyn | 0 | 0 | 0 | No | 0 | 0 | 80 | 0,0 | No | Enabl... | Enabl... | Yes | No | None
KMQ... | Normal | PermDyn | 0 | 0 | 0 | No | 0 | 0 | 80 | 0,0 | No | Enabl... | Enabl... | Yes | No | None
ATDIV... | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0,0 | No | Enabl... | Enabl... | Yes | No | None
ATDIV... | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0,0 | No | Enabl... | Enabl... | Yes | No | None
KMQ... | Normal | PermDyn | 0 | 0 | 0 | No | 0 | 0 | 80 | 0,0 | No | Enabl... | Enabl... | Yes | No | None
CSQ1:BA01:MQESA
Hub Time: Do, 03/16/2006 07:57 PM
Server Available
Queue Statistics - 9.212.128.31 - WOLF1

Figure 3-26 The Queue Statistics workspace

Depending on how the application environment is set up, the queues may or may not be application specific.

For the Smart Bank showcase, the following rules were implemented:

- System-focused situations
  - The existence of the monitored queue

    Monitoring does not help if the monitored object no longer exists. The product enables an alarm for all the erased queue objects. Once a queue is discovered, it monitors the existence of the queue for 24 hours.
  - Monitoring the relative queue depth

    The system's well-being requires that no queue gets full. A full queue may lead to messages crowding the dead letter queue, which may in turn lead to the channels stopping. This means that a local problem may spread across system boundaries.

    In order to be informed earlier on, a two-level escalation in severity is implemented. A warning event is raised when 70% of the queue is full, and a critical event is raised if 85% of the queue is full.

- Application-focused situations
  - For specific queues, a special watch is implemented. Even if the queue is not becoming full, a few applications require that the queue is swept up at short intervals. Meanwhile, in order to avoid getting an alarm when a new message has arrived, a situation that watches the latency (the period of inactivity) is set up. This is shown in Figure 3-27.

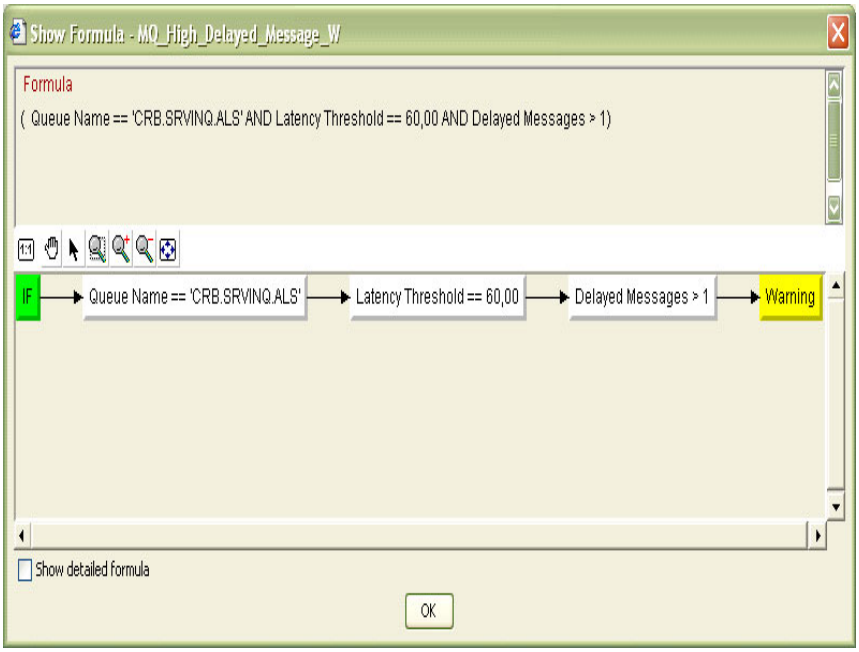


Figure 3-27 Situation MQ_High_Delayed_Messages_W

In this situation, the queue name is mentioned, as shown in Figure 3-27. This means that this rule applies to one specific queue only. This type of monitoring must be attached to a single queue at any time. For each queue requiring this type of monitoring, a single situation must be set up.

  - In order to view the applications, the according queue must always be open for reading. If the queue is not open, it indicates that the consuming program is not up and running correctly for some reason. This situation is useful for testing whether the trigger programs are correctly set up. When the MQ monitor is unable to test the existing processes, the information required is made available in this manner. Testing the existence of processes is performed through the operating system agent.

Note: There only a few examples available on WebSphere MQ Monitoring. The product comes with a lot of predefined situations. Most of them are not activated after the installation. We recommend that you review these rules and amend them to suit your installation.

3.3 IBM CICS Business Event Publisher for MQ Series and event processing

This section describes the way we used the CICS Business Event Publisher for MQ Series to populate our data warehouse database, in the following order:

- Overview of CICS Business Event Publisher for MQSeries
- The way this was implemented in our Smart Bank showcase infrastructure
- Physical views from the OMEGAMON XE for MQ to explain the CICS Business Event Publisher in action

3.3.1 Overview of CICS Business Event Publisher for MQ Series

CICS Business Event Publisher (CICS BEP) for MQSeries V1.2 helps you to gain maximum value out of your existing core business systems. It enables you to create MQ Series messages based on events within your applications, allowing you to rapidly integrate and extend the existing current CICS, DB2, and Information Management System (IMS) applications and data without the necessity to change application code.

The key features of CICS Business Event Publisher are:

- Detection of application and data events without changes being made to the underlying applications
- New approach that enables you to utilize the legacy applications in order to leverage new business opportunities or new technology
- Distribution of CICS, DB2, and IMS/DB information using MQ Series messages without rewriting code
- Delivery of external event logging or notification of CICS, DB2, or IMS activity
- Extension of data from the mainframe to other databases more efficiently

CICS Business Event Publisher for MQSeries helps monitor the events within the CICS Transaction Server, DB2, and IMS/DB. When certain events occur, for example, a file update request, CICS Business Event Publisher for MQSeries compares the attributes and data associated with the event against a set of rules you have defined. If a rule matches the event, it creates a message and sends it to an MQSeries message queue. By capturing critical business process events and creating messages from the event's associated data, you can quickly and effectively enhance and extend the existing applications without changing any application code.

By acting as an event publisher, CICS Business Event Publisher for MQSeries uses push technology, a simple, noninvasive capability that transports messages based on events that occur in the normal processing of the existing CICS, DB2, and IMS applications and data. The existing applications do not have to be modified, which in turn, significantly reduces the risk of disrupting the existing operations across the enterprise.

Through real-time monitoring, an entity that is external to the source data system can be notified about a change in the data held in a Virtual Storage Access Method (VSAM) record, DB2 table, or IMS segment. Based on the data change, the notification can also be used to trigger further activities. External event logging or notification about data change activity enables you to perform code logging functions without any application code in real time.

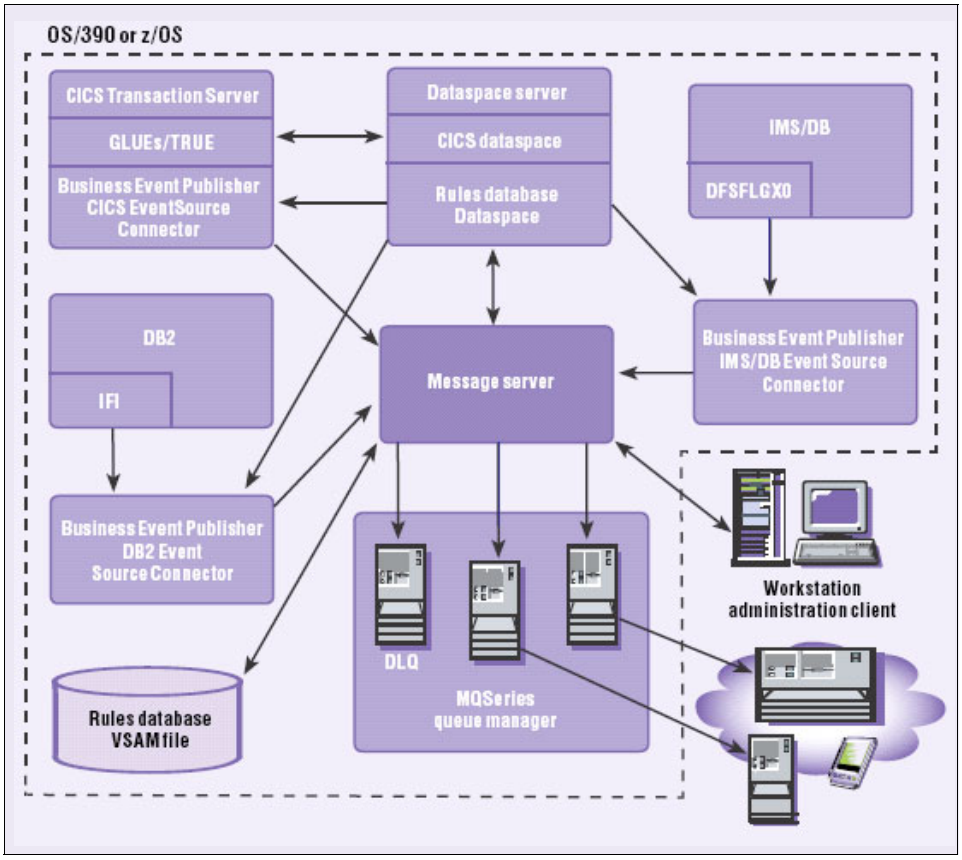


Figure 3-28 Overview of CICS Business Event Publisher for MQSeries 1.2

As shown in Figure 3-28, CICS Business Event Publisher comprises five components:

- CICS Business Event Publisher for MQSeries event connectors
- CICS Business Event Publisher for MQSeries message server
- CICS Business Event Publisher for MQSeries data space server
- CICS Business Event Publisher for MQSeries rules database
- The workstation administration client

CICS Business Event Publisher for MQSeries event connectors

CICS Business Event Publisher for MQSeries event connectors allow you to monitor events that occur within specific areas, including the following:

- A CICS event source connector running in a CICS region, which allows you to monitor VSAM file control, temporary storage, transient data, interval control, and program control (LINK) requests
- A DB2 event source connector running on a z/OS address space, which allows you to capture DB2 database events for any DB2 table with change data capture enabled

Note: In our installation, we use the event monitoring for DB2 to track the changes in the posting table in the banking database. Because every banking transaction involves an update of the posting table, the table is a good candidate for event monitoring.

- An IMS event source connector running in a z/OS address space that allows you to capture IMS/DB events for any database that has IMS change data capture enabled. This includes IMS/DB/TM, Database Control (DBCTL), and IMS batch. CICS Business Event Publisher for MQSeries supports a range of database types that include hierarchical indexed sequential access method (HISAM), simple hierarchical indexed sequential access method (SHISAM), hierarchical direct access method (HDAM), partitioned HDAM (PHDAM), hierarchical indexed direct access method (HIDAM), partitioned HIDAM PHIDAM, and data entry database (DEDB).

The connector determines whether a rule matches a particular event based on the selection criteria you have provided. When it finds a matching rule, the connector creates a message based on the message layout properties contained in the rule. It then passes the message and other rule properties to the message server through a highly efficient, cross-memory interface for subsequent delivery to an MQSeries message queue.

CICS Business Event Publisher for MQSeries event connectors anticipate the possibility that an event may fit multiple rules. Any or all the properties of the rules, such as message destination or message construction, can differ from one rule to the next. A rule can also specify various MQSeries MQPUT options that are to be used, with the message written to the MQSeries message queue for you to extend applications in numerous ways without changing the code.

CICS Business Event Publisher for MQSeries message server

The CICS Business Event Publisher for MQSeries message server uses attributes associated with a selected event, along with the rule you provide, to send a message created by the CICS Business Event Publisher event connector to the appropriate MQSeries message queue. While maintaining the integrity of the associated unit of work, the message server supports the use of multiple subtasks to process multiple units of work in parallel, providing a high level of performance. If the unit of work encompassing the event is not completed successfully, the event's message is *not* published. For DB2 and IMS connectors, events are processed asynchronously after the actual DB2 or IMS unit of work is complete. If a rule specifies that a message must participate in the unit of work, the message is not completed on the queue unless the unit of work being processed is successfully committed. The message server also contains a TCP/IP listener subtask used for communication between the message server and the Business Event Publisher workstation administration client. It can, for example, upload new rules created on the workstation administration client to the message server rules file.

CICS Business Event Publisher for MQSeries data space server

CICS Business Event Publisher for MQSeries data space server is used to provide virtual storage for semi-persistent information used by message servers and event source connectors. This allows message servers and event source connectors to be stopped and started without losing valuable state information, such as cached rules and CICS transaction pattern knowledge.

CICS Business Event Publisher for MQSeries rules database

CICS Business Event Publisher for MQSeries rules database is used as a repository for rules that contain the selection criteria imposed on the monitored events, and provides the data selection and layout criteria for the resulting message. Rules also contain the destination message queue names and MQSeries MQPUT options, along with properties that control subsequent processing, for example, when a message will be sent and when an action will be taken if an unrecoverable error is detected.

The rules are generated on a workstation and sent to the CICS Business Event Publisher for MQSeries message server using TCP/IP communication. Although they are physically stored in a VSAM file, these rules are ultimately cached in a data space owned by the data space server for efficient access and persistence across message server and connector restarts. CICS Business Event Publisher for MQSeries provides the necessary means to refresh the cache.

The rules database resides on the z/OS system and contains data built through a Windows technology-based graphical user interface (GUI). Refreshing the rules database is simple. You can make changes without interrupting the message server or losing work. If a refresh action takes place when a unit of work is in progress, it continues to use the previous rules database until it reaches a syncpoint. The rules database and workstation administration client utilize an easy-to-use hierarchy structure in which rules are contained in groups, and groups are contained in lists.

The CICS Business Event Publisher for MQSeries workstation administration client

The CICS Business Event Publisher for MQSeries workstation administration client is a Windows technology-based GUI used to create and maintain rules. After the rules are created, they are uploaded to the message server to be stored in the rules file. These rules can be downloaded later to a workstation, and modified and uploaded to the message server again. Rules provide a link between the Business Event Publisher event source connectors and the message server. The workstation administration client also contains a robust facility to generate rules and manipulate the contents of MQSeries messages generated by the rules. A utility is provided to import a Common Business Oriented Language (COBOL) copybook into the Business Event Publisher environment so that the selection criteria and message creation are based on named items.

Configure the workstation administration client to ensure that sensitive information is not left on the workstation after the information is uploaded to the message server. Besides helping you overcome shortcomings with regard to storage of rules, this also helps avoid a potential security threat.

3.3.2 Using CICS Business Event Publisher to populate the banking data warehouse database

In our Smart Bank showcase infrastructure, we monitor the events that occur on the posting table from the banking DB2 database. For each banking transaction a record is created in the posting table. We use this record event to populate our banking database warehouse database, which is located on another DB2 for z/OS subsystem.

Note: In our case, we want to populate a database located on the same IBM System z machine in the same logical partition. As mentioned earlier, we can use CICS Business Event Publisher to propagate the event to an external system. However, in this scenario, our intent is to only demonstrate the functionality from the CICS Business Event Publisher.

As specified in 3.3.1, “Overview of CICS Business Event Publisher for MQ Series” on page 154, we picked the relevant components for our infrastructure. Following are the installed components:

- CICS Business Event Publisher event connectors for DB2 on z/OS
- CICS Business Event Publisher message server
- CICS Business Event Publisher data space server
- The workstation administration client

Figure 3-29 shows the real implementation that was performed while writing this book.

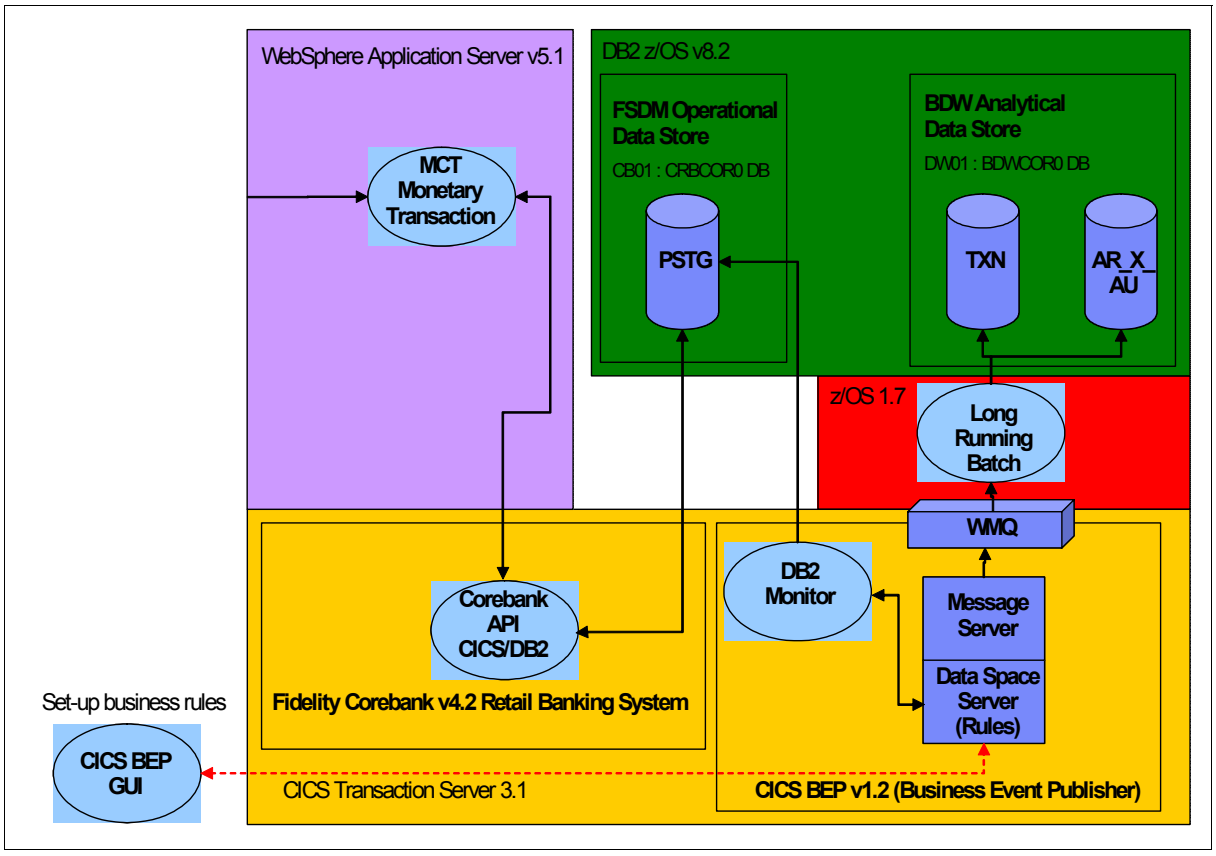


Figure 3-29 CICS Business Event Publisher as implemented in the Smart Bank showcase

The purpose of this book is *not* to detail the CICS Business Event Publisher setup process. For more details about this, refer to *CICS Business Event Publisher for MQSeries V1.2 Getting Started*, GC34-6296-002.

In order to understand the functioning of our setup, Table 3-1 provides the correspondence between component functions and the started tasks implemented for this book.

Table 3-1 CICS Business Event Publisher: Components and associated started tasks

CICS Business Event Publisher component	Implemented started task
Event collector for DB2	CBMDES1
DataSpace Server	CBMDSSVR
Messaging Server	CBMMSVR1

Our posting table, CRBCOR0.PSTG, is set up to track changes by setting the parameter DATA CAPTURE CHANGES in DB2. As soon as all the three CICS Business Event Publisher components are started, every change to the Smart Bank showcase posting table is reported and a message sent to the MQ queue, CSQ1.

In addition, a batch process to read our MQ queue, CSQ1, is started. Every message posted to the MQ queue is then analyzed and processed. At this juncture, two operations take place:

- Checking whether a banking client account exists, and if it does not, creating a record to the BDW.TXN table. An example of the SQL statements is shown in Example 3-1.

Example 3-1 Creating a record when an account does not exist

```
EXEC SQL SELECT *
INTO   :DCLAR-X-AU
FROM BDWCORO.AR_X_AU
WHERE  AR_ID = :WOO-ARF-IDFR
END-EXEC.
MOVE SQLCODE TO SQL-CODE.
DISPLAY 'SQLCODE = SEL  >' , DISPLAY-FIELD.

IF SQLCODE = 1OO AND TEST1 = 'OK'
   EXEC SQL INSERT INTO BDWCORO.AR_X_AU
   (  AR_ID ,
      AU_ID  ,
      AR_X_AU_TP_ID ,
      EFF_DT  ,
      PPN_DT  ,
      PPN_TM  ,
      END_DT )
    VALUES
    ( :WOO-ARF-IDFR,
      OOO1 ,
      3139 ,
      '2028-03-31' ,
      '2028-03-31' ,
      :ZZTIME ,
      '2028-03-31'      ) END-EXEC
```

- Updating BDW.TXN by adding a transaction to the specific account. Example 3-2 shows the INSERT statement performed by our batch process.

Example 3-2 Example of INSERT

```
EXEC SQL INSERT INTO BDWCORO.TXN
   ( TXN_TP_ID  ,
     TXN_ID ,
     AU_ID   ,
     PRD_BAL_TP_ID ,
     PNT_BAL_TP_ID ,
     MSR_PRD_ID ,
     UOM_ID ,
     CNSL_MTH_ID ,
     SRO_ID ,
     NET_CASH_FLOW_AMT ,
     PST_DT ,
     ACG_EFF_TP_ID ,
```

```
     TXN_DT  ,
     TXN_BOOK_DT ,
PPN_DT ,
        TXN_VAL_DT ,
        PPN_TM ,
        EXCP_CAUS_ID ,
        PST_ENTR_AMT ,
        EXCP_IMP_ID ,
        UNQ_ID_SRC_STM ,
        DVC_ID  )
       VALUES
       ( :TXN-TP-ID ,
         :WOO-TRANS-IDFR,
         :AU-ID1  ,
         :PRD-BAL-TP-ID ,
         :PNT-BAL-TP-ID ,
         :MSR-PRD-ID ,
         :WOO-CRNCY-IDFR-PSTG-AC ,
         :CNSL-MTH-ID ,
         :SRO-ID ,
         :WOO-ACCTG-AMT ,
         '2028-03-31' ,
         :ACG-EFF-TP-ID ,
         '2028-03-31' ,
         '2028-03-31' ,
        :REAL-DATE ,
         '2028-03-31' ,
         :ZZTIME ,
         :EXCP-CAUS-ID ,
         :WOO-ACCTG-BSE-AMT ,
         :EXCP-IMP-ID ,
         :UNQ-ID-SRC-STM ,
      :WOO-WS-ID  ) END-EXEC.
```

To perform an ongoing analysis of an incoming MQ message, a batch job, FABBEP1, is run. It uses the program CORBEP1, which is linked, edited, and then bound on DB2 earlier on. Example 3-3 shows a sample bind job for the CORBEP1 program.

Example 3-3 Sample bind job

```
//BEPBIND JOB AX4328,
//         CLASS=A,MSGCLASS=0,MSGLEVEL=(1,1),
//         NOTIFY=&SYSUID,
//         REGION=0M
/*JOBPARM SYSAFF*
//*********************************************************************
//*********************************************************************
//JOBLIB   DD  DISP=SHR,
//             DSN=DSN810.SDSNLOAD
//BIND     EXEC PGM=IKJEFT01,DYNAMNBR=20
//DBRMLIB  DD DISP=SHR,
//         DSN=DSNDW00.DBRMLIB.DATA
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
```

```
//SYSTSIN  DD  *
  DSN SYSTEM(DW01)
   BIND PACKAGE (CORBEP1) MEMBER(CORBEP1) ISO(CS) OW(DB2ADM) ACT(REP)
   BIND PLAN(CORBEP1) ISO(CS) -
             PKLIST(CORBEP1.CORBEP) OW(DB2ADM) ACT(REP)
```

Example 3-4 shows an example of the batch process.

Example 3-4 Example of batch process

```
//FABBEP1 JOB AX4328,
//         CLASS=A,MSGCLASS=0,MSGLEVEL=(1,1),
//         NOTIFY=&SYSUID,
//         REGION=4M
/*JOBPARM SYSAFF=*
//FABRUN  EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//DBRMLIB  DD DSN=DSNDW00.DBRMLIB.DATA,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//REPORT   DD SYSOUT=*
//SYSTSIN DD *
 DSN SYSTEM(DW01)
 RUN  PROGRAM(CORBEP1) PLAN(CORBEP1) -
      LIB('FABRICE.MQ.USER.LOAD') -
      PARMS('CSQ1,CSQ1,CICBEP.QLOCAL1,1,A,1000,X')
 END
//*
```

3.3.3 Monitoring a running CICS Business Event Publisher using OMEGAMON XE for MQSeries

This section provides details about monitoring the running CICS Business Event Publisher process, especially the activity on the MQ queue.

1. In our environment, several machines are monitored using IBM Tivoli Monitoring V6.1. When you log in to the TEP, a login window is displayed. In the top left corner of the TEP window, navigate and expand the navigator tree to locate the MQSeries queue.

   Note: In our example, queue CSQ1 is installed on the z/OS LPAR named BA01. The activity on CSQ1 is monitored.

Figure 3-30 shows the expanded TEP navigator tree, and the information provided for the CSQ1 queue. This figure shows that several other monitoring agents are installed in our environment. These monitor other subsystems such as DB2, CICS, mainframe network, sysplex, and so on. The queue manager summary provides the status of the queue manager that is active.

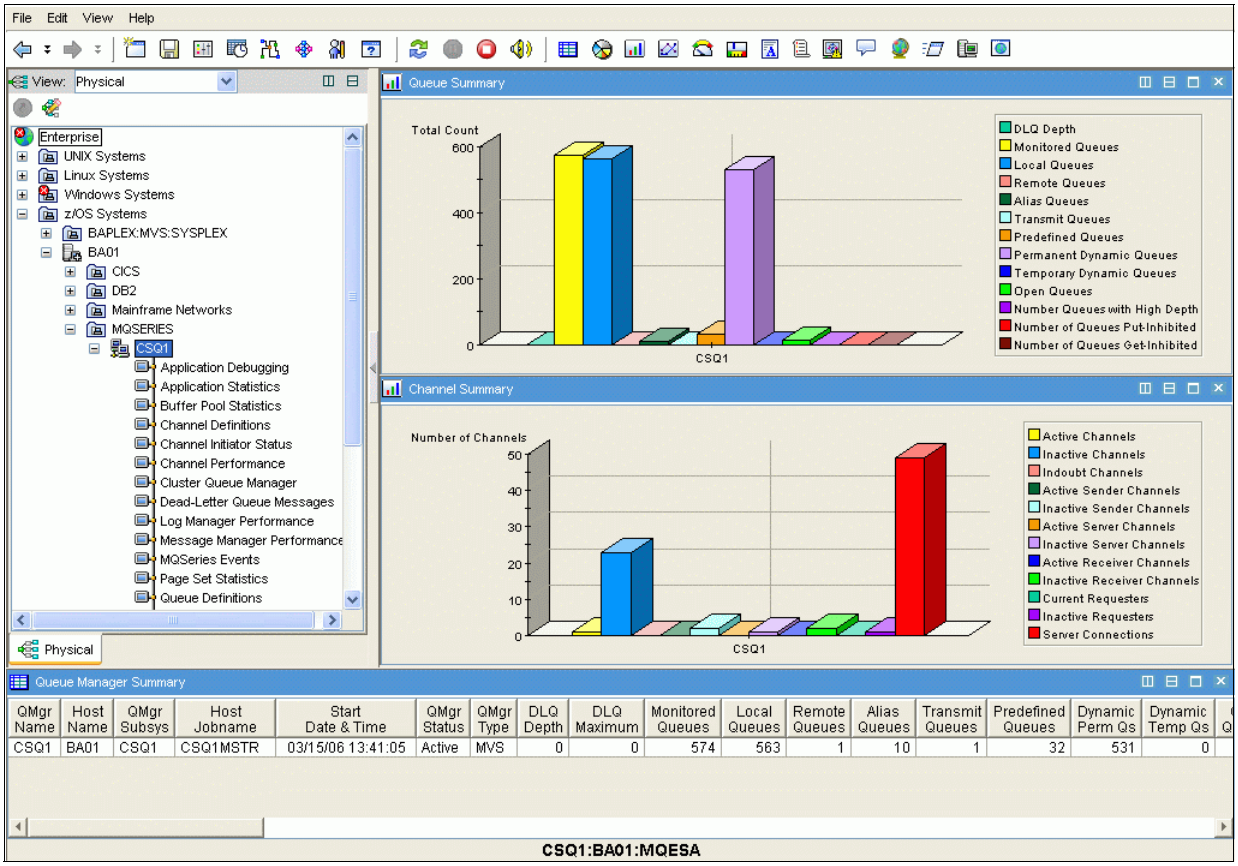


Figure 3-30 Navigating to MQSeries queue information

2. Let us now look at the queue definitions. To access more details about the CICS Business Event Publisher activity, click the **Queue Definitions** tab. The Queue Definitions Summary is displayed, as shown in Figure 3-31. Several queues are defined in this window. The queue called CICBEP.QLOCAL1 is the one we look at in our case. It is defined as a local queue. Drill down to this particular queue by clicking the blue anchor next to the queue name. Before performing this action, in order to view the queue statistics, click the **Queue Statistics** tab in the Navigator view.

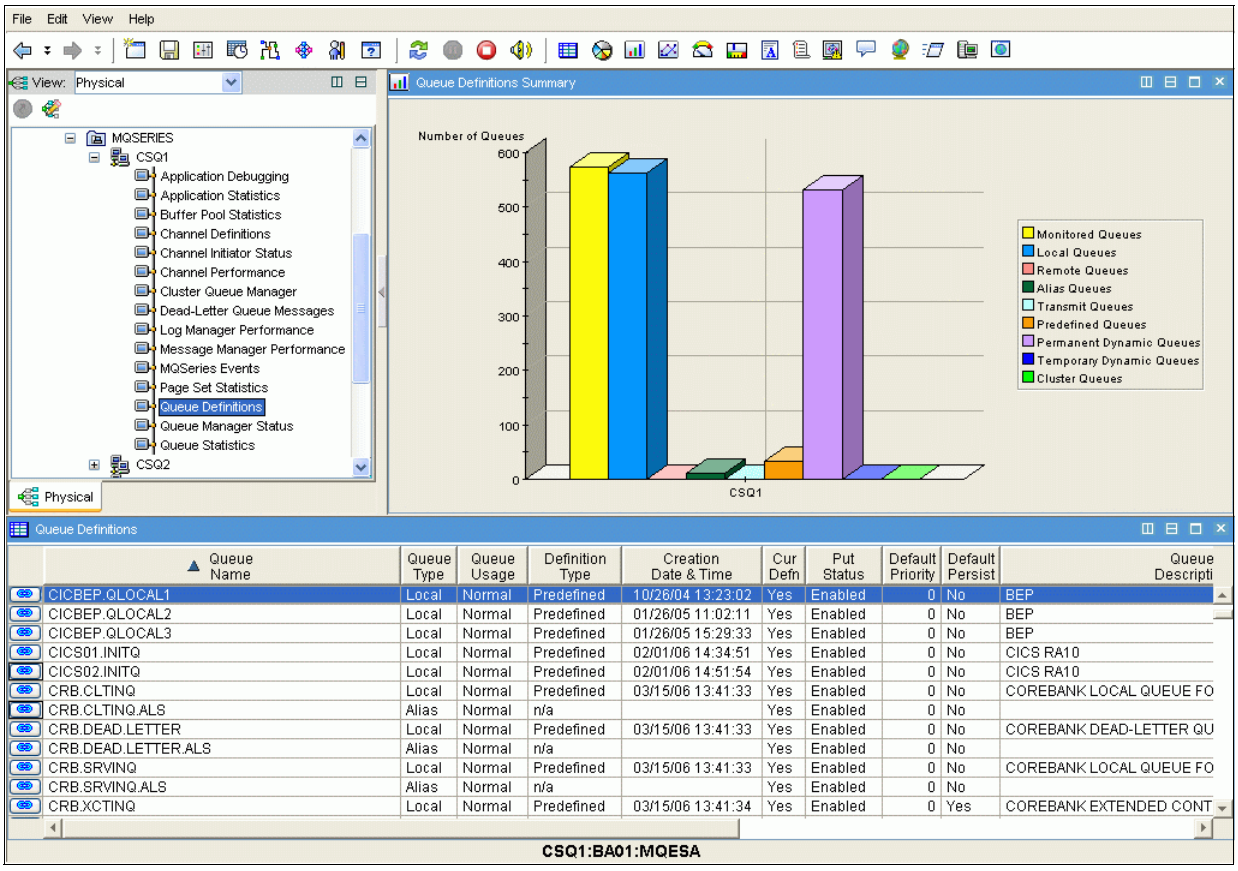

File Edit View Help
View: Physical
MQSERIES
CSQ1
Application Debugging
Application Statistics
Buffer Pool Statistics
Channel Definitions
Channel Initiator Status
Channel Performance
Cluster Queue Manager
Dead-Letter Queue Messages
Log Manager Performance
Message Manager Performance
MQSeries Events
Page Set Statistics
Queue Definitions
Queue Manager Status
Queue Statistics
CSQ2
Physical
Queue Definitions Summary
Number of Queues
600
500
400
300
200
100
0
CSQ1
Monitored Queues
Local Queues
Remote Queues
Alias Queues
Transmit Queues
Predefined Queues
Permanent Dynamic Queues
Temporary Dynamic Queues
Cluster Queues
Queue Definitions
Queue Name | Queue Type | Queue Usage | Definition Type | Creation Date & Time | Cur Defn | Put Status | Default Priority | Default Persist | Queue Descripti
CICBEP.QLOCAL1 | Local | Normal | Predefined | 10/26/04 13:23:02 | Yes | Enabled | 0 | No | BEP
CICBEP.QLOCAL2 | Local | Normal | Predefined | 01/26/05 11:02:11 | Yes | Enabled | 0 | No | BEP
CICBEP.QLOCAL3 | Local | Normal | Predefined | 01/26/05 15:29:33 | Yes | Enabled | 0 | No | BEP
CICS01.INITQ | Local | Normal | Predefined | 02/01/06 14:34:51 | Yes | Enabled | 0 | No | CICS RA10
CICS02.INITQ | Local | Normal | Predefined | 02/01/06 14:51:54 | Yes | Enabled | 0 | No | CICS RA10
CRB.CLTINQ | Local | Normal | Predefined | 03/15/06 13:41:33 | Yes | Enabled | 0 | No | COREBANK LOCAL QUEUE FO
CRB.CLTINQ.ALS | Alias | Normal | n/a | | Yes | Enabled | 0 | No |
CRB.DEAD.LETTER | Local | Normal | Predefined | 03/15/06 13:41:33 | Yes | Enabled | 0 | No | COREBANK DEAD-LETTER QU
CRB.DEAD.LETTER.ALS | Alias | Normal | n/a | | Yes | Enabled | 0 | No |
CRB.SRVINQ | Local | Normal | Predefined | 03/15/06 13:41:33 | Yes | Enabled | 0 | No | COREBANK LOCAL QUEUE FO
CRB.SRVINQ.ALS | Alias | Normal | n/a | | Yes | Enabled | 0 | No |
CRB.XCTINQ | Local | Normal | Predefined | 03/15/06 13:41:34 | Yes | Enabled | 0 | Yes | COREBANK EXTENDED CONT
CSQ1:BA01:MQESA

Figure 3-31 Queue definitions

3. The Queue Statistics Summary is displayed, as shown in Figure 3-32. This window shows that there is currently no activity on the CICBEP.QLOCAL1 queue. All the metrics related to the queue activity, such as Total Opens, Input Opens, Output Opens, Current Depth, % Full, are set to zero. Before starting any workload on this queue, drill down one more time to view the statistics information pertaining to CICBEP.QLOCAL1. Click the blue anchor next to the queue name.

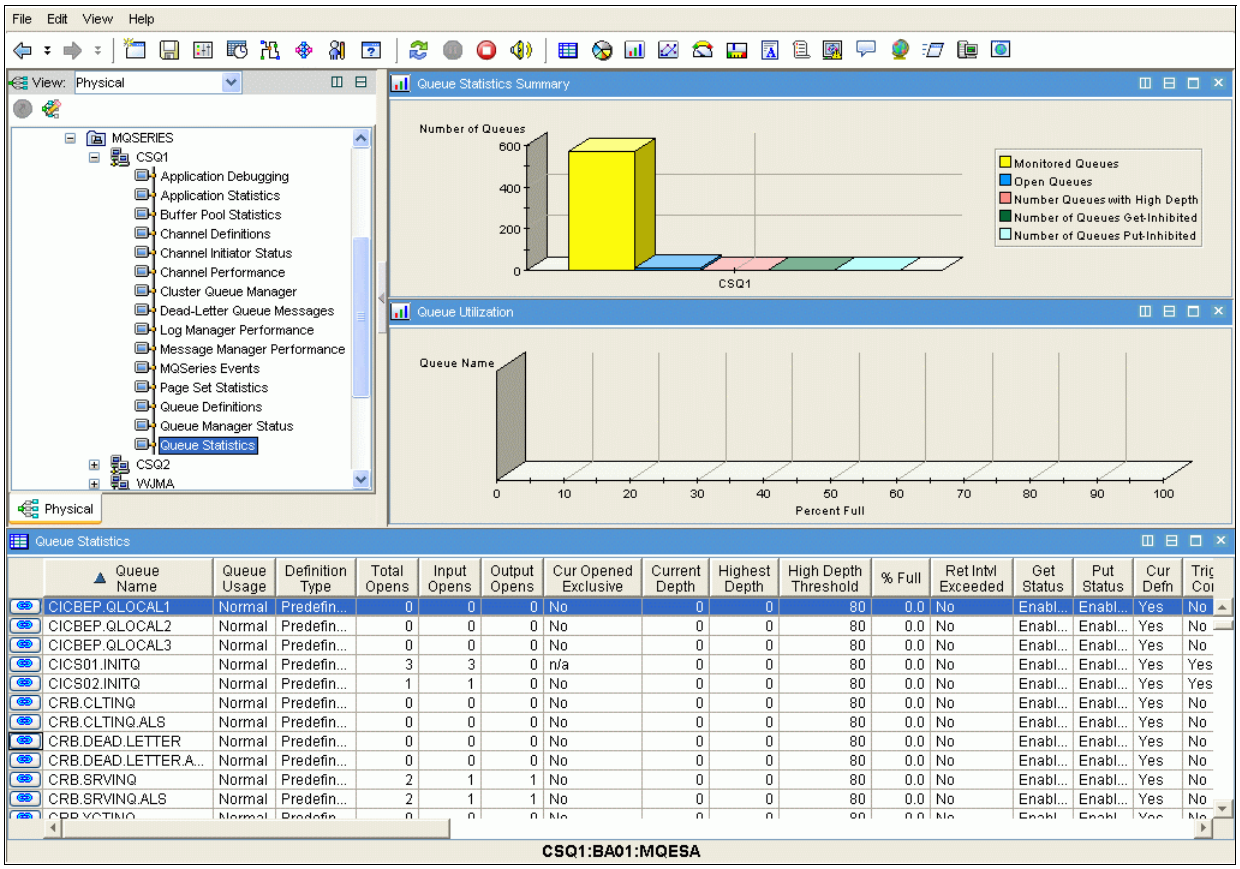

File Edit View Help
View: Physical
MQSERIES
CSQ1
Application Debugging
Application Statistics
Buffer Pool Statistics
Channel Definitions
Channel Initiator Status
Channel Performance
Cluster Queue Manager
Dead-Letter Queue Messages
Log Manager Performance
Message Manager Performance
MQSeries Events
Page Set Statistics
Queue Definitions
Queue Manager Status
Queue Statistics
CSQ2
WJMA
Physical
Queue Statistics Summary
Number of Queues
600
400
200
0
CSQ1
Monitored Queues
Open Queues
Number Queues with High Depth
Number of Queues Get-Inhibited
Number of Queues Put-Inhibited
Queue Utilization
Queue Name
0 10 20 30 40 50 60 70 80 90 100
Percent Full
Queue Statistics
Queue Name | Queue Usage | Definition Type | Total Opens | Input Opens | Output Opens | Cur Opened Exclusive | Current Depth | Highest Depth | High Depth Threshold | % Full | Ret Intvl Exceeded | Get Status | Put Status | Cur Defn | Trig Col
CICBEP.QLOCAL1 | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CICBEP.QLOCAL2 | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CICBEP.QLOCAL3 | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CICS01.INITQ | Normal | Predefin... | 3 | 3 | 0 | n/a | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | Yes
CICS02.INITQ | Normal | Predefin... | 1 | 1 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | Yes
CRB.CLTINQ | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CRB.CLTINQ.ALS | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CRB.DEAD.LETTER | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CRB.DEAD.LETTER.A... | Normal | Predefin... | 0 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CRB.SRVINQ | Normal | Predefin... | 2 | 1 | 1 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CRB.SRVINQ.ALS | Normal | Predefin... | 2 | 1 | 1 | No | 0 | 0 | 80 | 0.0 | No | Enabl... | Enabl... | Yes | No
CSQ1:BA01:MQESA

Figure 3-32 Queue statistics

4. The window shown in Figure 3-33 is displayed. The Recent Queue Statistics panel at the bottom shows the latest statistics for each interval (1 minute is our default interval duration).

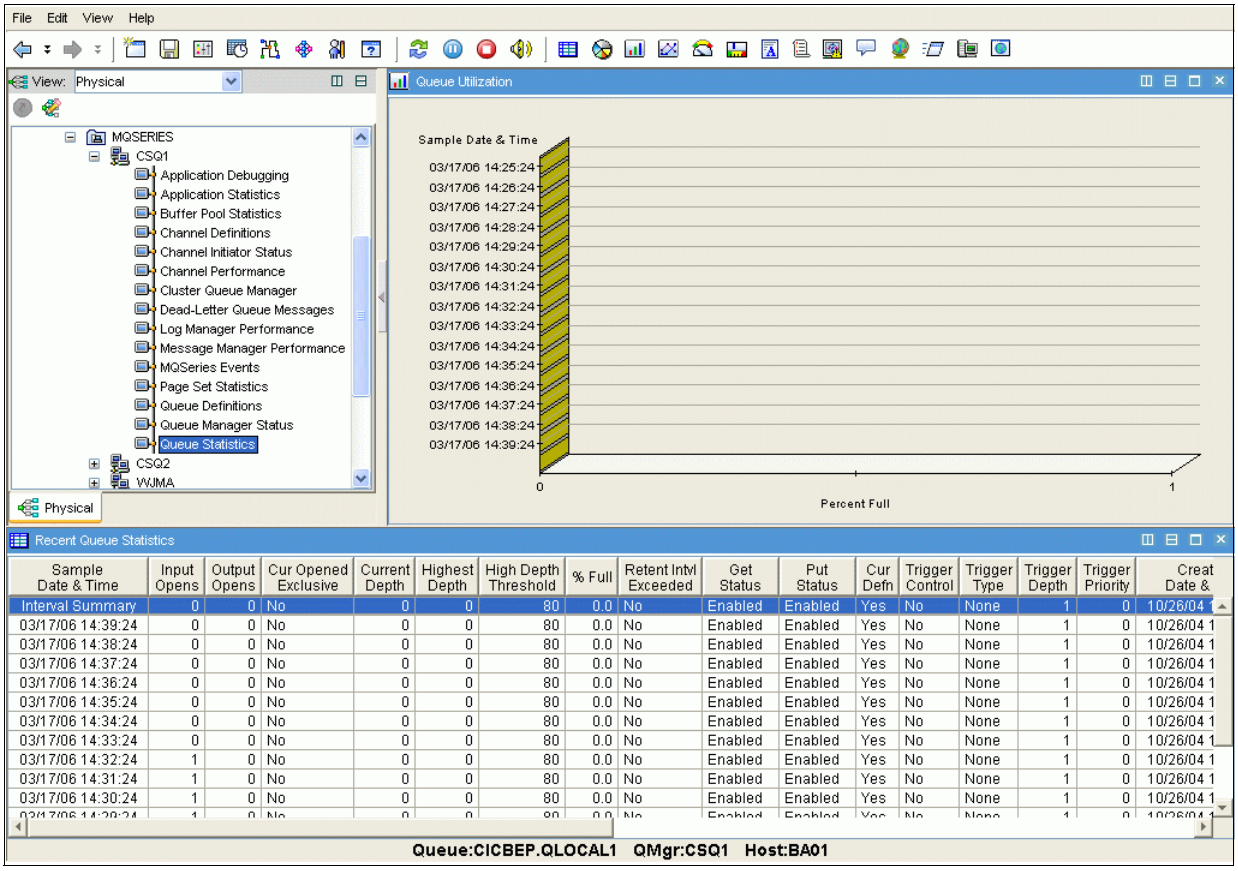

File Edit View Help
View: Physical
MQSERIES
CSQ1
Application Debugging
Application Statistics
Buffer Pool Statistics
Channel Definitions
Channel Initiator Status
Channel Performance
Cluster Queue Manager
Dead-Letter Queue Messages
Log Manager Performance
Message Manager Performance
MQSeries Events
Page Set Statistics
Queue Definitions
Queue Manager Status
Queue Statistics
CSQ2
WJMA
Physical
Queue Utilization
Sample Date & Time
03/17/06 14:25:24
03/17/06 14:26:24
03/17/06 14:27:24
03/17/06 14:28:24
03/17/06 14:29:24
03/17/06 14:30:24
03/17/06 14:31:24
03/17/06 14:32:24
03/17/06 14:33:24
03/17/06 14:34:24
03/17/06 14:35:24
03/17/06 14:36:24
03/17/06 14:37:24
03/17/06 14:38:24
03/17/06 14:39:24
0
1
Percent Full
Recent Queue Statistics
Sample Date & Time | Input Opens | Output Opens | Cur Opened Exclusive | Current Depth | Highest Depth | High Depth Threshold | % Full | Retent Intvl Exceeded | Get Status | Put Status | Cur Defn | Trigger Control | Trigger Type | Trigger Depth | Trigger Priority | Creat Date &
Interval Summary | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:39:24 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:38:24 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:37:24 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:36:24 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:35:24 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:34:24 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:33:24 | 0 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:32:24 | 1 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:31:24 | 1 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
03/17/06 14:30:24 | 1 | 0 | No | 0 | 0 | 80 | 0.0 | No | Enabled | Enabled | Yes | No | None | 1 | 0 | 10/26/04
Queue:CICBEP.QLOCAL1 QMgr:CSQ1 Host:BA01

Figure 3-33 Queue CICBEP.QLOCAL1 statistics

To demonstrate the activity on the MQ queue, perform the following tasks:

a. Start the CICS Business Event Publisher-started tasks.

b. Using a workload generator, create banking transactions on the banking infrastructure. As stated earlier, for each money transaction carried out (which records a change in the DB2 posting table), the DB2 change is tracked and a message is written to the queue. To see how the MQ queue is filling up, the batch FABBEP (to process messages in the queue) is not started.

   Figure 3-34 shows that the following new metrics have changed:

   - Output Opens is at 19 according to the CICS Business Event Publisher process that is running
   - Current Depth increases. The rate is about 300 messages inserted in the queue every minute. When this is compared to the transaction rate as seen from the workload generator, it matches our rate of 5.3 transactions per second.

- % Full increases. We deliberately undersized our queue to have some displayable information. The maximum number of messages that can be written on the queue set to 8000.

The queue utilization graph shown in Figure 3-34, reflects this activity on the queue.

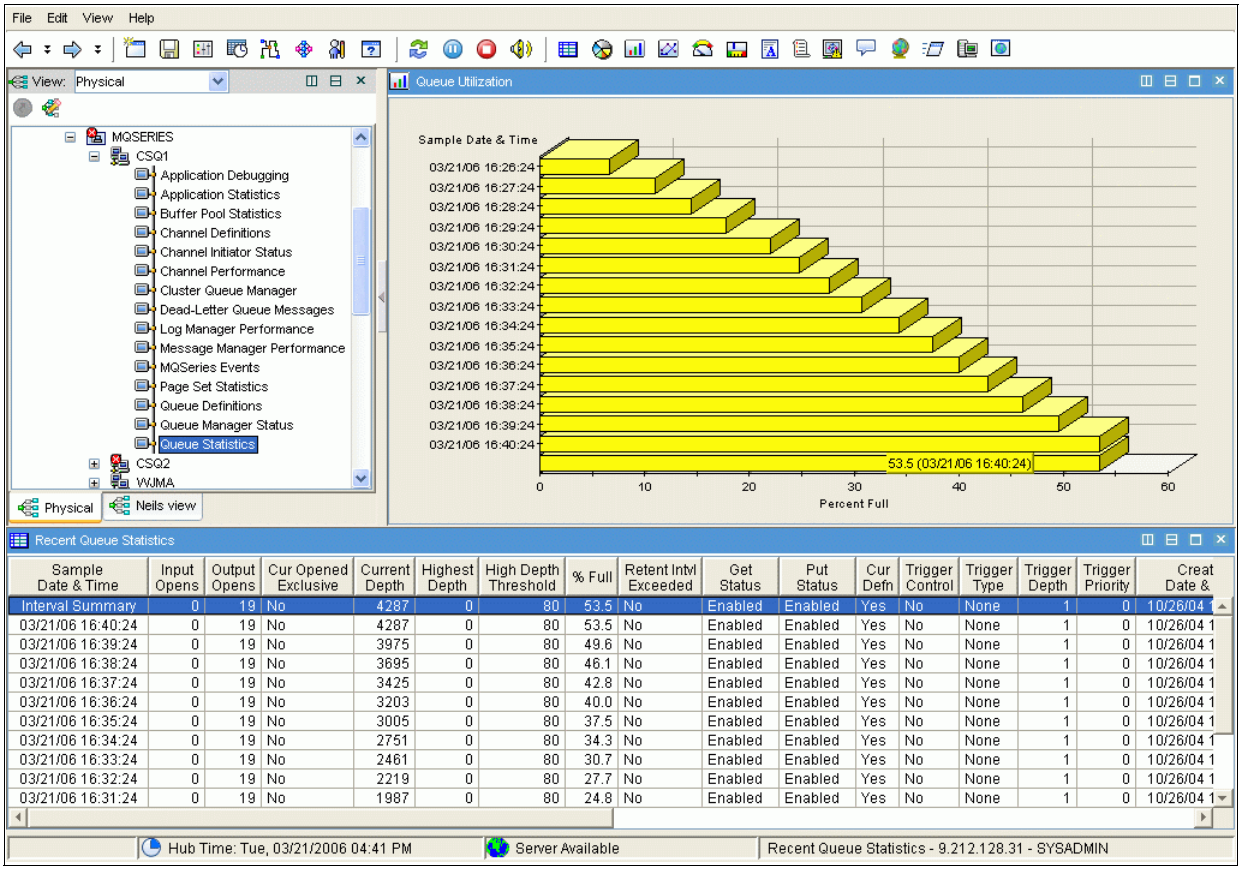

File Edit View Help
View: Physical
MQSERIES
CSQ1
Application Debugging
Application Statistics
Buffer Pool Statistics
Channel Definitions
Channel Initiator Status
Channel Performance
Cluster Queue Manager
Dead-Letter Queue Messages
Log Manager Performance
Message Manager Performance
MQSeries Events
Page Set Statistics
Queue Definitions
Queue Manager Status
Queue Statistics
CSQ2
WJMA
Physical
Neils view
Queue Utilization
Sample Date & Time
03/21/06 16:26:24
03/21/06 16:27:24
03/21/06 16:28:24
03/21/06 16:29:24
03/21/06 16:30:24
03/21/06 16:31:24
03/21/06 16:32:24
03/21/06 16:33:24
03/21/06 16:34:24
03/21/06 16:35:24
03/21/06 16:36:24
03/21/06 16:37:24
03/21/06 16:38:24
03/21/06 16:39:24
03/21/06 16:40:24
53.5 (03/21/06 16:40:24)
0 10 20 30 40 50 60
Percent Full
Recent Queue Statistics
Sample Date & Time | Input Opens | Output Opens | Cur Opened Exclusive | Current Depth | Highest Depth | High Depth Threshold | % Full | Retent Intvl Exceeded | Get Status | Put Status | Cur Defn | Trigger Control | Trigger Type | Trigger Depth | Trigger Priority | Creat Date &
Interval Summary 0 19 No 4287 0 80 53.5 No Enabled Enabled Yes No None 1 0 10/26/04
03/21/06 16:40:24 0 19 No 4287 0 80 53.5 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:39:24 0 19 No 3975 0 80 49.6 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:38:24 0 19 No 3695 0 80 46.1 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:37:24 0 19 No 3425 0 80 42.8 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:36:24 0 19 No 3203 0 80 40.0 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:35:24 0 19 No 3005 0 80 37.5 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:34:24 0 19 No 2751 0 80 34.3 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:33:24 0 19 No 2461 0 80 30.7 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:32:24 0 19 No 2219 0 80 27.7 No Enabled Enabled Yes No None 1 0 10/26/04 1
03/21/06 16:31:24 0 19 No 1987 0 80 24.8 No Enabled Enabled Yes No None 1 0 10/26/04 1
Hub Time: Tue, 03/21/2006 04:41 PM
Server Available
Recent Queue Statistics - 9.212.128.31 - SYSADMIN

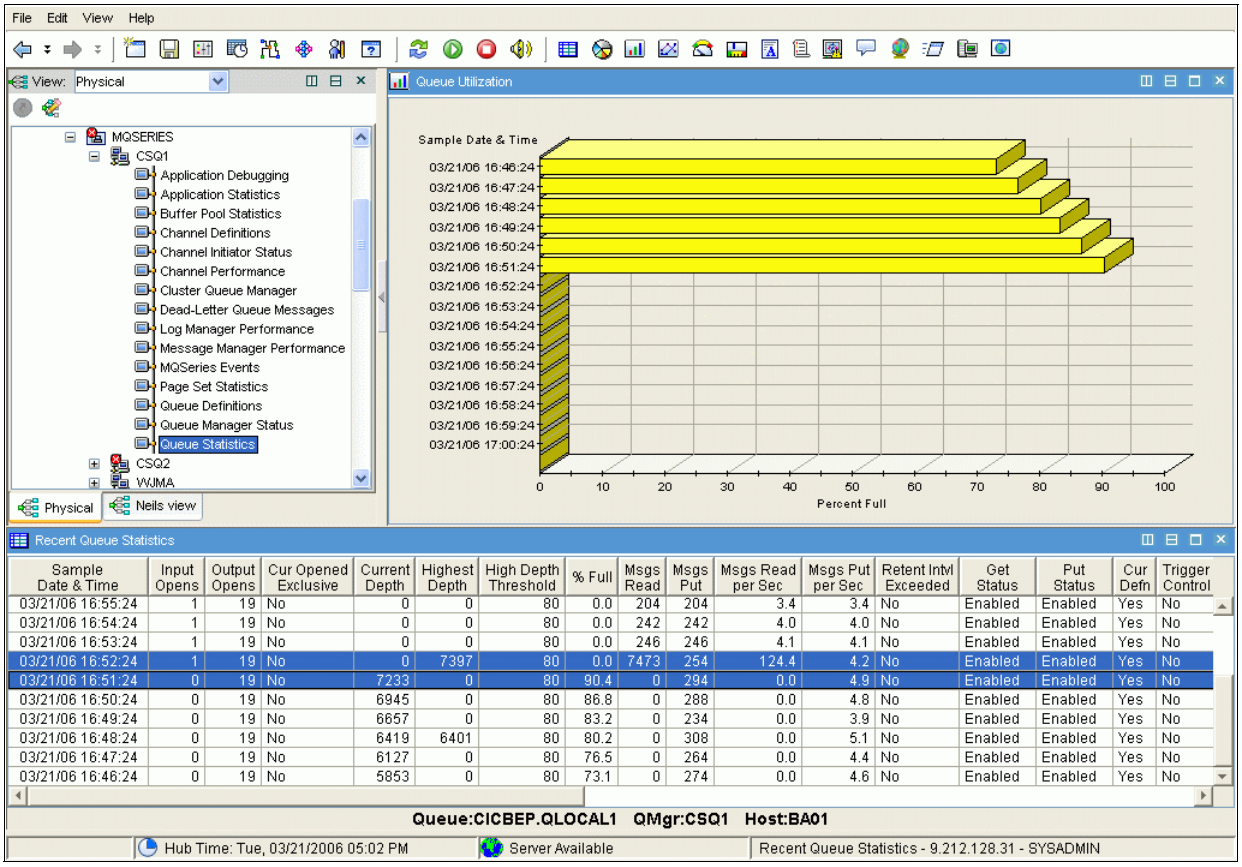
Figure 3-34 Queue CICBEP.QLOCAL1 filling up due to workload generation

c. Start the FABBEP batch to read all the messages that are currently in the queue. The messages will be processed and the DB2 posting table populated as described in 3.3.2, “Using CICS Business Event Publisher to populate the banking data warehouse database” on page 157.

The result on the TEP is shown in Figure 3-35. This figure shows that several new metrics have changed:

- Output Opens is still at 19 because the CICS Business Event Publisher process is still running
- Current Depth is now going from 7233 to 0 during one interval. All the messages present in the queue have been processed immediately after starting the FABBEPbatch
- % Full decreases, and is at 0
- MessagesRead is at 7423 messages, with a rate of 124.4 MessagesRead per second
- The writing to the queue activity is continued. Messages put is 254 messages during a one minute interval.

The queue utilization graph shown in Figure 3-35 reflects this processing activity of the queue. It is empty. The ongoing FABBEP batch reads all the new messages as soon as they arrive.



Sample Date & Time	Input Opens	Output Opens	Cur Opened Exclusive	Current Depth	Highest Depth	High Depth Threshold	% Full	Msgs Read	Msgs Put	Msgs Read per Sec	Msgs Put per Sec	Retent Intvl Exceeded	Get Status	Put Status	Cur Defn	Trigger Control
03/21/06 16:55:24	1	19	No	0	0	80	0.0	204	204	3.4	3.4	No	Enabled	Enabled	Yes	No
03/21/06 16:54:24	1	19	No	0	0	80	0.0	242	242	4.0	4.0	No	Enabled	Enabled	Yes	No
03/21/06 16:53:24	1	19	No	0	0	80	0.0	246	246	4.1	4.1	No	Enabled	Enabled	Yes	No
03/21/06 16:52:24	1	19	No	0	7397	80	0.0	7473	254	124.4	4.2	No	Enabled	Enabled	Yes	No
03/21/06 16:51:24	0	19	No	7233	0	80	90.4	0	294	0.0	4.9	No	Enabled	Enabled	Yes	No
03/21/06 16:50:24	0	19	No	6945	0	80	86.8	0	288	0.0	4.8	No	Enabled	Enabled	Yes	No
03/21/06 16:49:24	0	19	No	6657	0	80	83.2	0	234	0.0	3.9	No	Enabled	Enabled	Yes	No
03/21/06 16:48:24	0	19	No	6419	6401	80	80.2	0	308	0.0	5.1	No	Enabled	Enabled	Yes	No
03/21/06 16:47:24	0	19	No	6127	0	80	76.5	0	264	0.0	4.4	No	Enabled	Enabled	Yes	No
03/21/06 16:46:24	0	19	No	5853	0	80	73.1	0	274	0.0	4.6	No	Enabled	Enabled	Yes	No

Figure 3-35 Processing activity of the queue with the FABBEP batch started

5. Finally, in order to monitor what is happening on the DB2 subsystem, navigate to DW01:BA01:DB2, where DW01 is the subsystem ID of the DataWarehouse DB2, and BA01 is the logical partition (LPAR) name where this DB2 is running.

   Figure 3-36 shows the details pertaining to the current DB2 threads on the DW01 subsystem. A thread is open for plan CORBEP1. This plan has been allocated by the job FABBEP.

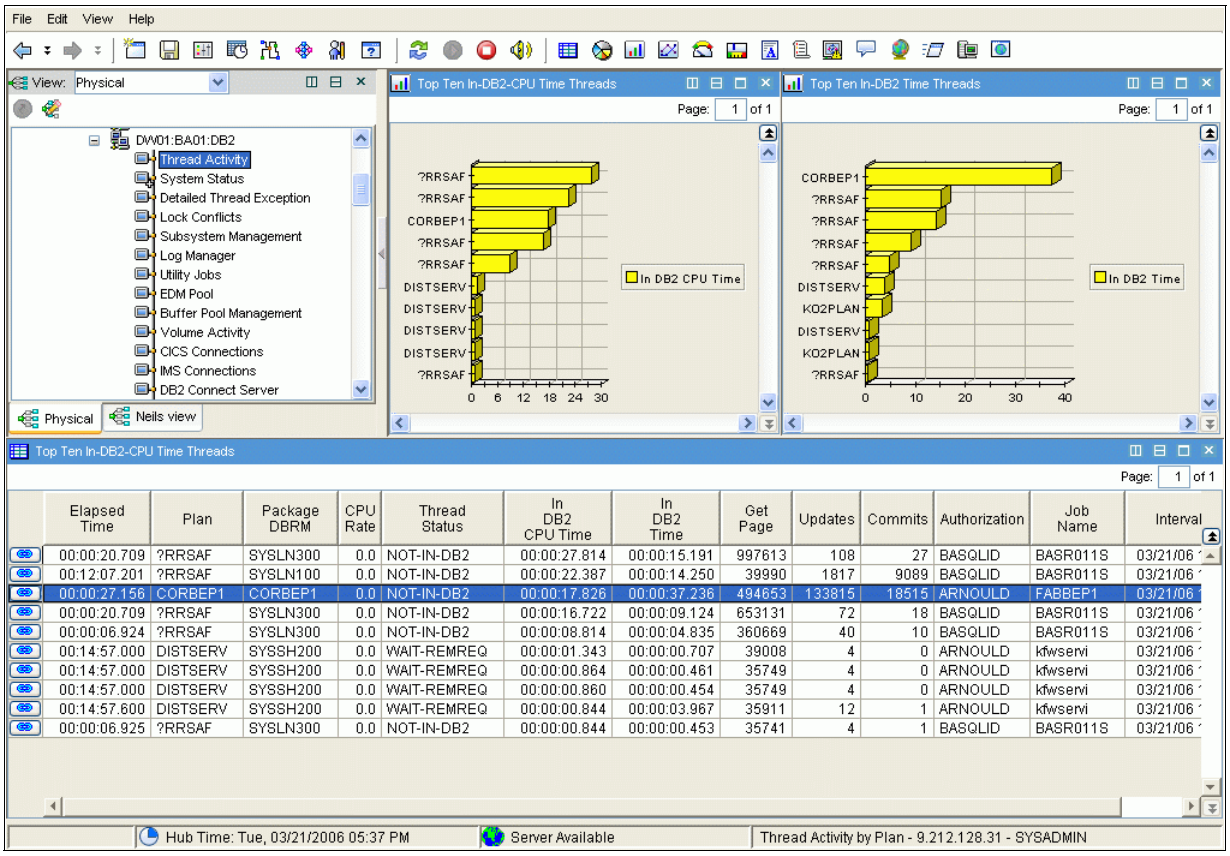

File Edit View Help
View: Physical
DW01:BA01:DB2
Thread Activity
System Status
Detailed Thread Exception
Lock Conflicts
Subsystem Management
Log Manager
Utility Jobs
EDM Pool
Buffer Pool Management
Volume Activity
CICS Connections
IMS Connections
DB2 Connect Server
Physical
Neils view
Top Ten In-DB2-CPU Time Threads
Page: 1 of 1
?RRSAF
?RRSAF
CORBEP1
?RRSAF
?RRSAF
DISTSERV
DISTSERV
DISTSERV
DISTSERV
?RRSAF
In DB2 CPU Time
0 6 12 18 24 30
Top Ten In-DB2 Time Threads
Page: 1 of 1
CORBEP1
?RRSAF
?RRSAF
?RRSAF
?RRSAF
DISTSERV
KO2PLAN
DISTSERV
KO2PLAN
?RRSAF
In DB2 Time
0 10 20 30 40
Top Ten In-DB2-CPU Time Threads
Page: 1 of 1
Elapsed Time | Plan | Package DBRM | CPU Rate | Thread Status | In DB2 CPU Time | In DB2 Time | Get Page | Updates | Commits | Authorization | Job Name | Interval
00:00:20.709 | ?RRSAF | SYSLN300 | 0.0 | NOT-IN-DB2 | 00:00:27.814 | 00:00:15.191 | 997613 | 108 | 27 | BASQLID | BASR011S | 03/21/06
00:12:07.201 | ?RRSAF | SYSLN100 | 0.0 | NOT-IN-DB2 | 00:00:22.387 | 00:00:14.250 | 39990 | 1817 | 9089 | BASQLID | BASR011S | 03/21/06
00:00:27.156 | CORBEP1 | CORBEP1 | 0.0 | NOT-IN-DB2 | 00:00:17.826 | 00:00:37.236 | 494653 | 133815 | 18515 | ARNOULD | FABBEP1 | 03/21/06
00:00:20.709 | ?RRSAF | SYSLN300 | 0.0 | NOT-IN-DB2 | 00:00:16.722 | 00:00:09.124 | 653131 | 72 | 18 | BASQLID | BASR011S | 03/21/06
00:00:06.924 | ?RRSAF | SYSLN300 | 0.0 | NOT-IN-DB2 | 00:00:08.814 | 00:00:04.835 | 360669 | 40 | 10 | BASQLID | BASR011S | 03/21/06
00:14:57.000 | DISTSERV | SYSSH200 | 0.0 | WAIT-REMREQ | 00:00:01.343 | 00:00:00.707 | 39008 | 4 | 0 | ARNOULD | kfwservi | 03/21/06
00:14:57.000 | DISTSERV | SYSSH200 | 0.0 | WAIT-REMREQ | 00:00:00.864 | 00:00:00.461 | 35749 | 4 | 0 | ARNOULD | kfwservi | 03/21/06
00:14:57.000 | DISTSERV | SYSSH200 | 0.0 | WAIT-REMREQ | 00:00:00.860 | 00:00:00.454 | 35749 | 4 | 0 | ARNOULD | kfwservi | 03/21/06
00:14:57.600 | DISTSERV | SYSSH200 | 0.0 | WAIT-REMREQ | 00:00:00.844 | 00:00:03.967 | 35911 | 12 | 1 | ARNOULD | kfwservi | 03/21/06
00:00:06.925 | ?RRSAF | SYSLN300 | 0.0 | NOT-IN-DB2 | 00:00:00.844 | 00:00:00.453 | 35741 | 4 | 1 | BASQLID | BASR011S | 03/21/06
Hub Time: Tue, 03/21/2006 05:37 PM
Server Available
Thread Activity by Plan - 9.212.128.31 - SYSADMIN

Figure 3-36 DB2 threads activity with the FABBEP job running

A plan called CBMPLAN, which is allocated by the CBMDES1-started task (the CICS Business Event Publisher event monitor-started task), is also found, as shown in Figure 3-37. On the thread line, click the blue anchor next to the time to get more details about this particular thread.

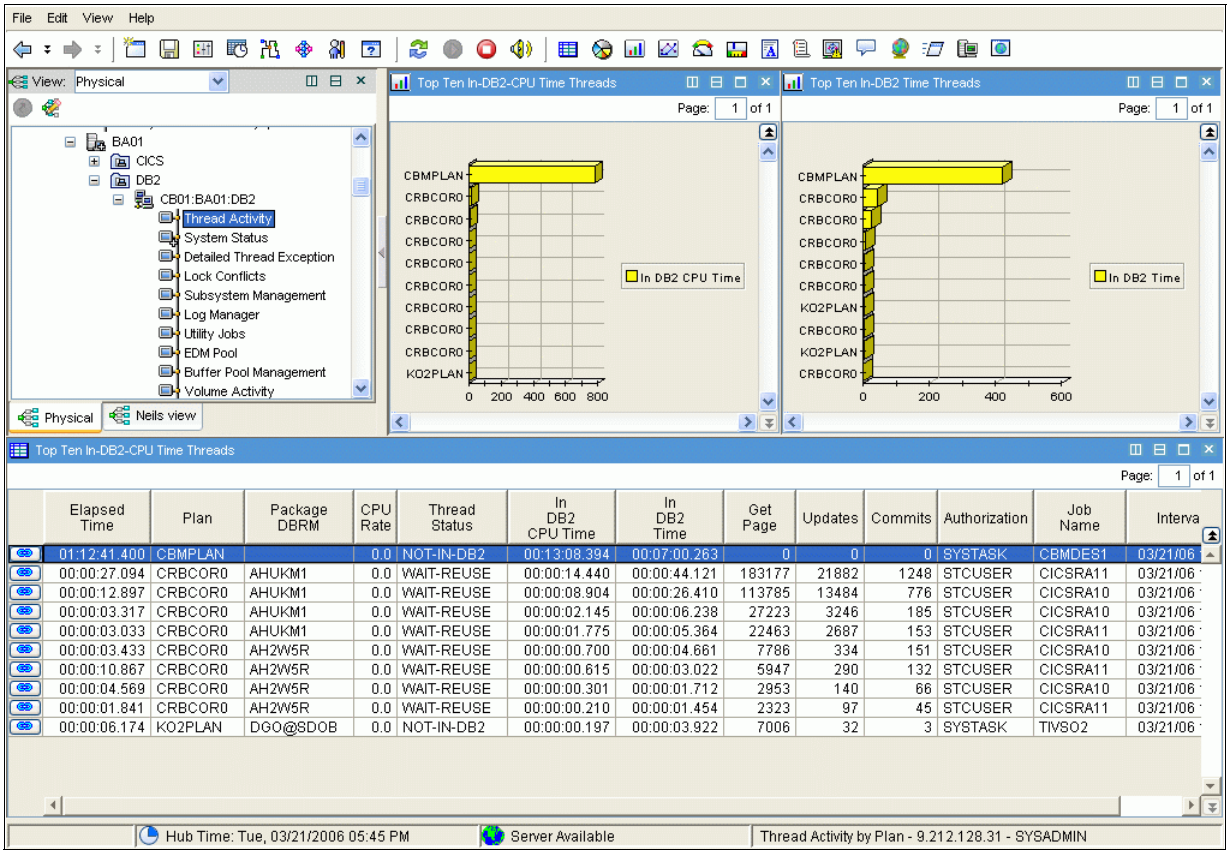

File Edit View Help
View: Physical
BA01
CICS
DB2
CB01:BA01:DB2
Thread Activity
System Status
Detailed Thread Exception
Lock Conflicts
Subsystem Management
Log Manager
Utility Jobs
EDM Pool
Buffer Pool Management
Volume Activity
Physical
Neils view
Top Ten In-DB2-CPU Time Threads
Page: 1 of 1
CBMPLAN
CRBCOR0
KO2PLAN
In DB2 CPU Time
0 200 400 600 800
Top Ten In-DB2 Time Threads
In DB2 Time
0 200 400 600
Top Ten In-DB2-CPU Time Threads
Elapsed Time | Plan | Package DBRM | CPU Rate | Thread Status | In DB2 CPU Time | In DB2 Time | Get Page | Updates | Commits | Authorization | Job Name | Interva
01:12:41.400 CBMPLAN 0.0 NOT-IN-DB2 00:13:08.394 00:07:00.263 0 0 0 SYSTASK CBMDES1 03/21/06
00:00:27.094 CRBCOR0 AHUKM1 0.0 WAIT-REUSE 00:00:14.440 00:00:44.121 183177 21882 1248 STCUSER CICSRA11 03/21/06
00:00:12.897 CRBCOR0 AHUKM1 0.0 WAIT-REUSE 00:00:08.904 00:00:26.410 113785 13484 776 STCUSER CICSRA10 03/21/06
00:00:03.317 CRBCOR0 AHUKM1 0.0 WAIT-REUSE 00:00:02.145 00:00:06.238 27223 3246 185 STCUSER CICSRA10 03/21/06
00:00:03.033 CRBCOR0 AHUKM1 0.0 WAIT-REUSE 00:00:01.775 00:00:05.364 22463 2687 153 STCUSER CICSRA11 03/21/06
00:00:03.433 CRBCOR0 AH2W5R 0.0 WAIT-REUSE 00:00:00.700 00:00:04.661 7786 334 151 STCUSER CICSRA10 03/21/06
00:00:10.867 CRBCOR0 AH2W5R 0.0 WAIT-REUSE 00:00:00.615 00:00:03.022 5947 290 132 STCUSER CICSRA11 03/21/06
00:00:04.569 CRBCOR0 AH2W5R 0.0 WAIT-REUSE 00:00:00.301 00:00:01.712 2953 140 66 STCUSER CICSRA10 03/21/06
00:00:01.841 CRBCOR0 AH2W5R 0.0 WAIT-REUSE 00:00:00.210 00:00:01.454 2323 97 45 STCUSER CICSRA11 03/21/06
00:00:06.174 KO2PLAN DGO@SDOB 0.0 NOT-IN-DB2 00:00:00.197 00:00:03.922 7006 32 3 SYSTASK TIVSO2 03/21/06
Hub Time: Tue, 03/21/2006 05:45 PM
Server Available
Thread Activity by Plan - 9.212.128.31 - SYSADMIN

Figure 3-37 DB2 thread activity with the CICS Business Event Publisher plan

6. The window shown in Figure 3-38 is displayed. Note that the CICS Business Event Publisher process is small. The CPU consumption and the elapsed time IN-DB2 are low. The plan is allocated and uses resources only when changes occur in the posting table. Because the MQ queue is monitored constantly, there are not many messages to process at a given time.

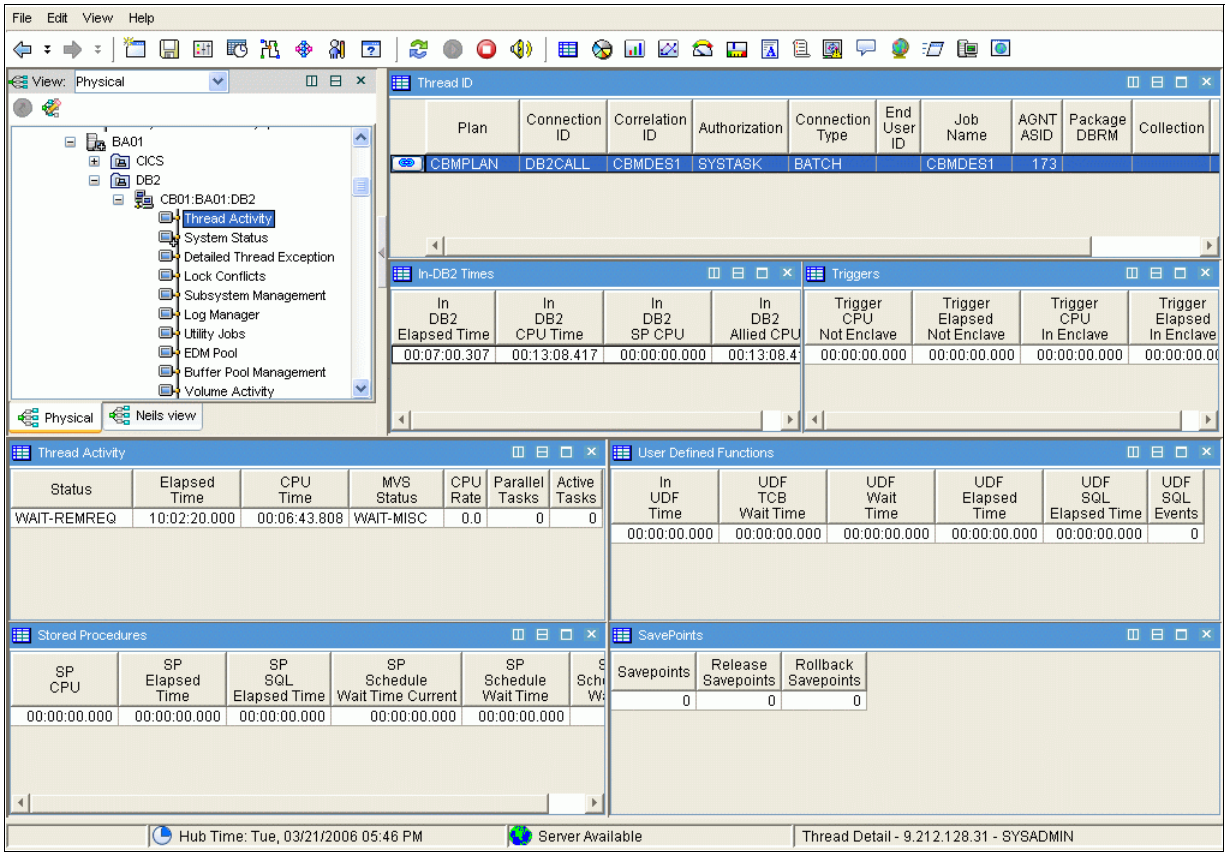


Figure 3-38 DB2 thread details for CBMPLAN

3.4 Multichannel architecture: Physical views

This section describes the Physical views that can be used in a multichannel architecture.

3.4.1 WebSphere Application Server

These Physical views are similar to those of the z/OS environment. Refer to 3.1.1, “IBM OMEGAMON for z/OS V3.10” on page 125 to obtain a broad view of IBM Tivoli Monitoring capabilities for WebSphere Application Server.

3.4.2 Uniform Resource Locator monitoring

The Hypertext Transfer Protocol (HTTP) data provider available in the Universal Agent enables you to obtain statistics regarding the Uniform Resource Locator (URL) connections initiated from a single location to miscellaneous Web sites. The installation and settings are described in 2.2, “The distributed environment” on page 84.

The content of the attribute groups relates to the settings defined during the configuration of the HTTP data provider.

The managed system <hostname>:INTERNET00 contains two attribute groups with data (the others are empty). The first one called MANAGED_URL contains a set of useful information pertaining to the URL, such as availability (status) and response time.

- Status

  When the remote server replies to the URL without any error, the URL has the status OK. The HTTP provider has the ability to manage the exceptions when the URL encounters problems such as:

  - The server does not exist, and the status field indicates “Not a recognized network address”
  - The HTTP service does not reply, and the status field indicates “URI authority not found”
  - The HTTP service is up, but the page does not exist, and the status field indicates “404: resource not found”

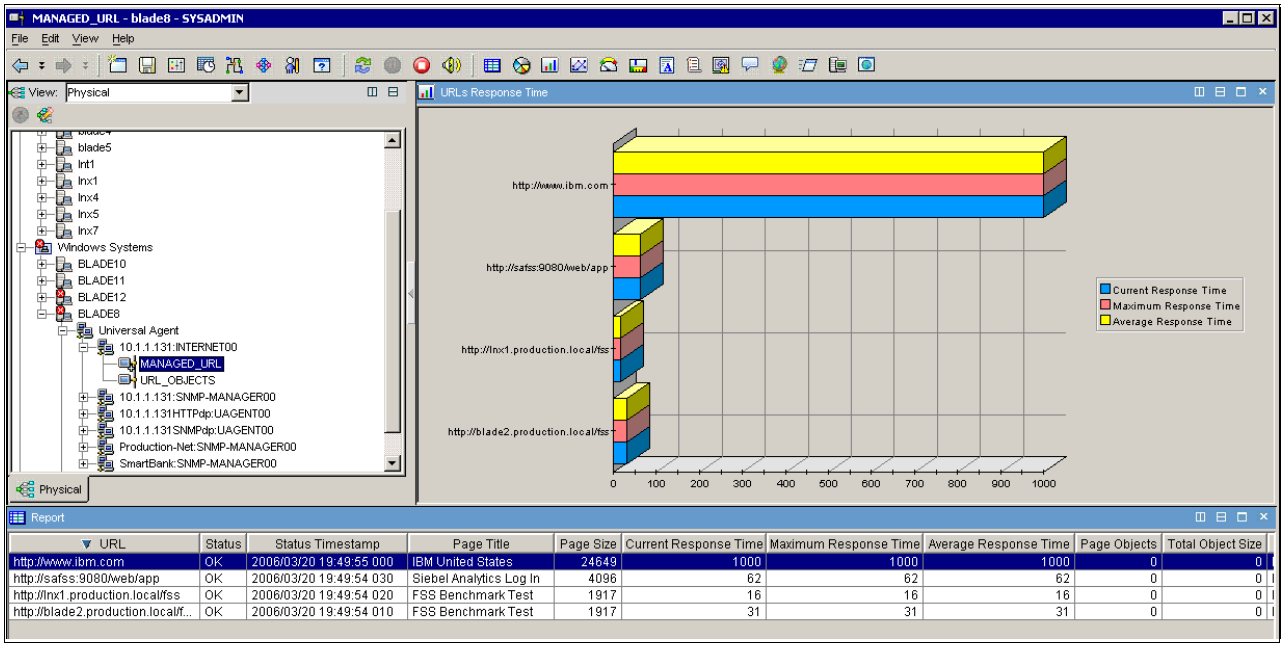
- Current response time

  The response time field specifies the number of milliseconds taken to load the content of a Web page. This value is a good way of reflecting the amount of time required by users when connecting for the first time to an URL (no data in browser’s cache). In such a situation, the Universal Agent must be installed in a location that is close to the users.

  The measurement can be mitigated, assuming that a part of the Web page is stored normally in the browser’s cache (the Universal Agent does not cache data). The percentage of objects that can be cached can be assessed as follows:

  - Look in the URL_OBJECTS workspace to identify the number and size of the miscellaneous objects such as GIF and JPEG images that must be cached by a browser. The MANAGED_URL workspace provides the total number of bytes downloaded for the objects in the Total Object Size field.
  - Estimating the percentage of objects that must be cached (50% means that if the page contains ten objects, only five will be downloaded by the HTTP provider to estimate the response time). Generally, content such as pictures, of a Web site is updated regularly. Consequently, a part of the objects may change. Thereby, the calculation of the cache percentage is only an assessment.
  - Reflecting this value in the configuration file KUMPURLS using the ObjCache% parameter

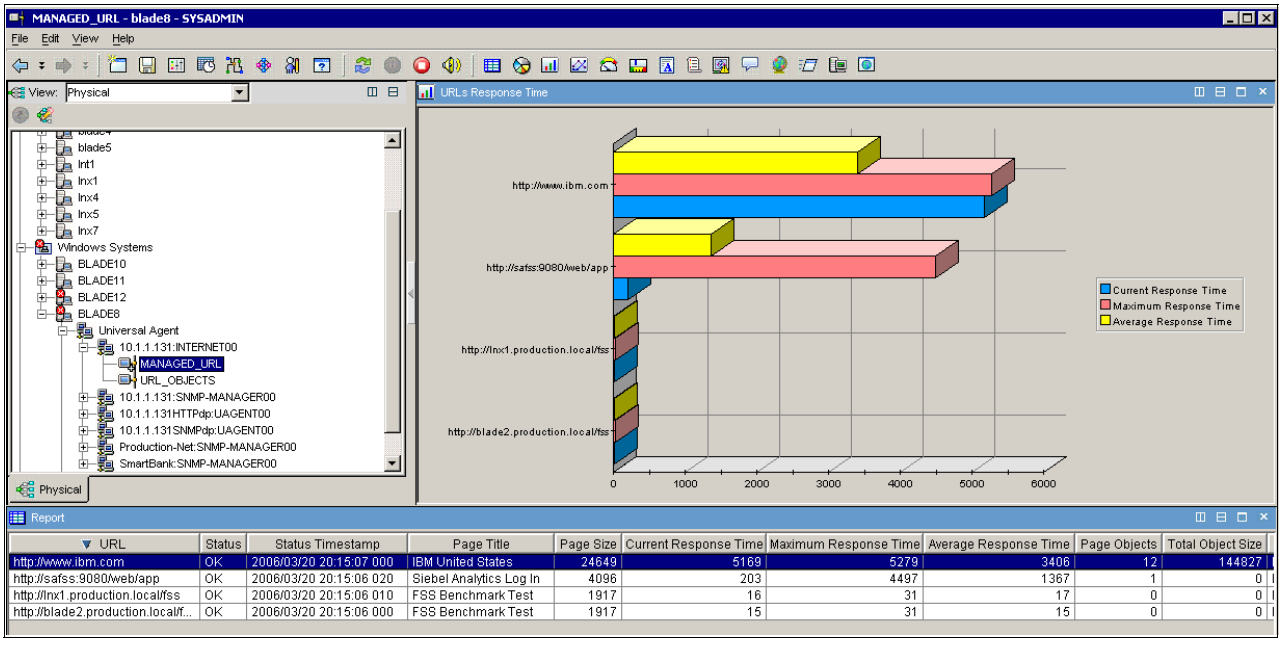
Figure 3-39 shows the URL's response time for the Web site www.ibm.com, with the parameter ObjCache%=100 (no object downloaded as in a situation when all of them are already in a cache). The response time is around 1 second.



URL	Status	Status Timestamp	Page Title	Page Size	Current Response Time	Maximum Response Time	Average Response Time	Page Objects	Total Object Size
http://www.ibm.com	OK	2006/03/20 19:49:55 000	IBM United States	24649	1000	1000	1000	0	0
http://safss:9080/web/app	OK	2006/03/20 19:49:54 030	Siebel Analytics Log In	4096	62	62	62	0	0
http://lnx1.production.local/fss	OK	2006/03/20 19:49:54 020	FSS Benchmark Test	1917	16	16	16	0	0
http://blade2.production.local/f...	OK	2006/03/20 19:49:54 010	FSS Benchmark Test	1917	31	31	31	0	0

Figure 3-39 TEP: Internet URL response time with ObjCache

Figure 3-40 shows the URL's response time for the site www.ibm.com with the parameter ObjCache%=0 (all objects are downloaded). The response time is around 5 seconds.



URL	Status	Status Timestamp	Page Title	Page Size	Current Response Time	Maximum Response Time	Average Response Time	Page Objects	Total Object Size
http://www.ibm.com	OK	2006/03/20 20:15:07 000	IBM United States	24649	5169	5279	3406	12	144827
http://safss:9080/web/app	OK	2006/03/20 20:15:06 020	Siebel Analytics Log In	4096	203	4497	1367	1	0
http://lnx1.production.local/fss	OK	2006/03/20 20:15:06 010	FSS Benchmark Test	1917	16	31	17	0	0
http://blade2.production.local/f...	OK	2006/03/20 20:15:06 000	FSS Benchmark Test	1917	15	31	15	0	0

Figure 3-40 TEP: Internet URL response time with no ObjCache

Note: The ObjCache parameter may be considered when monitoring URLs containing objects, such as JPEG and GIF images, in order to reflect a more realistic value. The response time is impacted by the total objects' size, especially for connections passing through low bandwidth network.

- Average and maximum response time

  The default sampling interval to check the URL's availability and response time is 300 seconds. However, this may differ for each URL, based on the parameter StatusInterval specified in the KUMPURLS file. The calculation for the average and maximum response time is always based on the last 15 minutes of sampling data. For long-term analysis, the historical data collection must be enabled.

  Figure 3-41 shows the maximum and average response time based on the last 15 minutes for miscellaneous URLs.

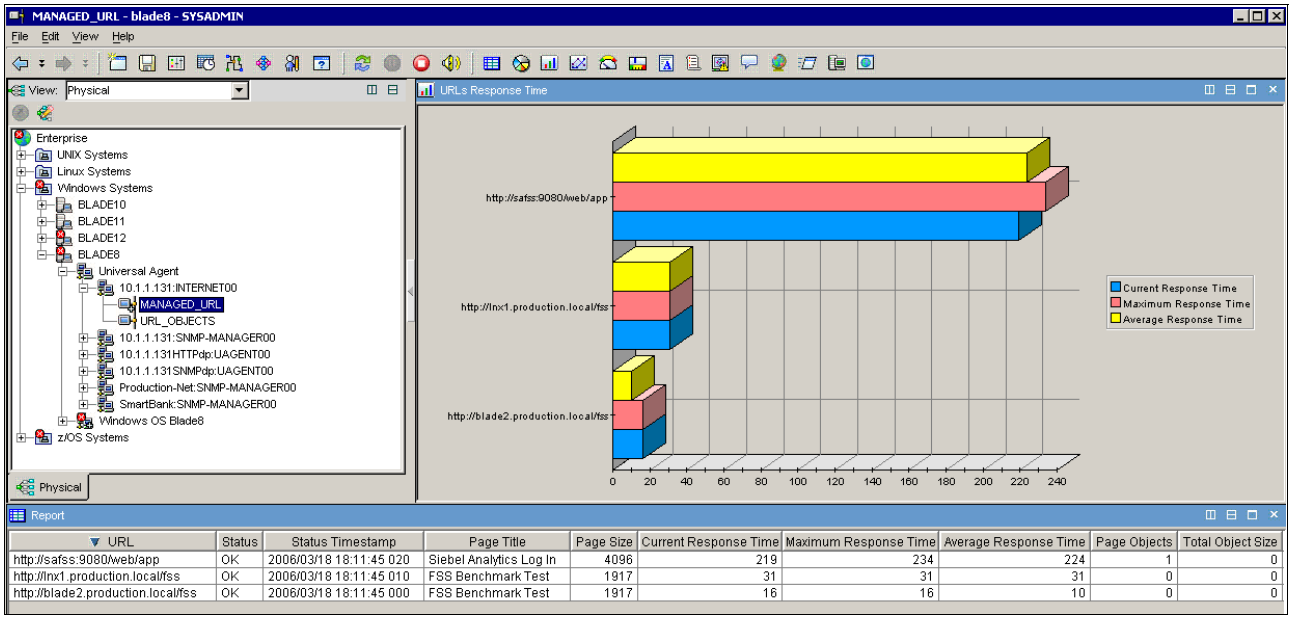

MANAGED_URL - blade8 - SYSADMIN
File Edit View Help
View: Physical
Enterprise
UNIX Systems
Linux Systems
Windows Systems
BLADE10
BLADE11
BLADE12
BLADE8
Universal Agent
10.1.1.131:INTERNET00
MANAGED_URL
URL_OBJECTS
10.1.1.131:SNMP-MANAGER00
10.1.1.131HTTPdp:UAGENT00
10.1.1.131SNMPdp:UAGENT00
Production-Net:SNMP-MANAGER00
SmartBank:SNMP-MANAGER00
Windows OS Blade8
z/OS Systems
Physical
URLs Response Time
http://safss:9080/web/app
http://lnx1.production.local/fss
http://blade2.production.local/fss
0 20 40 60 80 100 120 140 160 180 200 220 240
Current Response Time
Maximum Response Time
Average Response Time
Report
URL | Status | Status Timestamp | Page Title | Page Size | Current Response Time | Maximum Response Time | Average Response Time | Page Objects | Total Object Size
http://safss:9080/web/app | OK | 2006/03/18 18:11:45 020 | Siebel Analytics Log In | 4096 | 219 | 234 | 224 | 1 | 0
http://lnx1.production.local/fss | OK | 2006/03/18 18:11:45 010 | FSS Benchmark Test | 1917 | 31 | 31 | 31 | 0 | 0
http://blade2.production.local/fss | OK | 2006/03/18 18:11:45 000 | FSS Benchmark Test | 1917 | 16 | 16 | 10 | 0 | 0

Figure 3-41 TEP: MANAGED_URL workspace

3.4.3 IBM HTTP Server monitoring

The file data provider available in the Universal Agent enables the extraction of useful information from sequential text files. In such a situation, this feature is used to retrieve significant data from the IBM HTTP Server's access_log file. The installation and settings are described in 2.2, "The distributed environment" on page 84.

The content of the attribute groups is related to the settings defined during the configuration of the file data provider.

The managed system <hostname>:IBMHTTP00 contains 15 attribute groups. These groups provide data about exceptions, workload, and connection statistics. This section describes only those relating to the exception and workload, including the following:

- The attribute group EXCEPTION_DETAIL provides information about failed sessions. This occurs when the server receives a request from a remote client, but encounters an exception and is unable to fulfill the request. The client receives an error message indicating the request has failed. Thereafter, only the most significant attributes listed here are detailed:
  - Client location attribute

    This is the IP address of the client (remote host) that makes the request to the server.
  - Service status attribute

    This attribute indicates the HTTP status code. This indication is useful to determine the origin of the exception (client or server).

    Following are the most common client errors:

    - 400 Bad Request
    - 401 Unauthorized
    - 402 Payment Required
    - 403 Forbidden
    - 404 Not Found
    - 405 Method Not Allowed
    - 406 Not Acceptable
    - 407 Proxy Authentication Required
    - 408 Request Timeout
    - 409 Conflict
    - 410 Gone
    - 411 Length Required
    - 412 Precondition Failed
    - 413 Request Entity Too Large
    - 414 Request URI Too Long
    - 415 Unsupported Media Type

    Following are the most common server errors:

    - 500 Internal Server Error
    - 501 Not Implemented
    - 502 Bad Gateway
    - 503 Service Unavailable
    - 504 Gateway Timeout
    - 505 HTTP Version Not Supported

    To obtain all the details pertaining to the error codes, refer to Request For Comment (RFC) 2616, which is available on the Web at:

    http://rfc.net/rfc2616.html

    In the context of client error, the actual severity and impact may be difficult to determine, for example, Error 404 means that the server has not found anything that matches the client request. This error may be due to a bad client request, that is, wrong URL, or due to a resource such as the Hypertext Markup Language (HTML) page being moved on the server, as a result of which a client can no longer access it.

– Request attribute

This contains the resource requested by the client. Figure 3-42 shows the errors detected in the access_log file of an IBM HTTP Server.

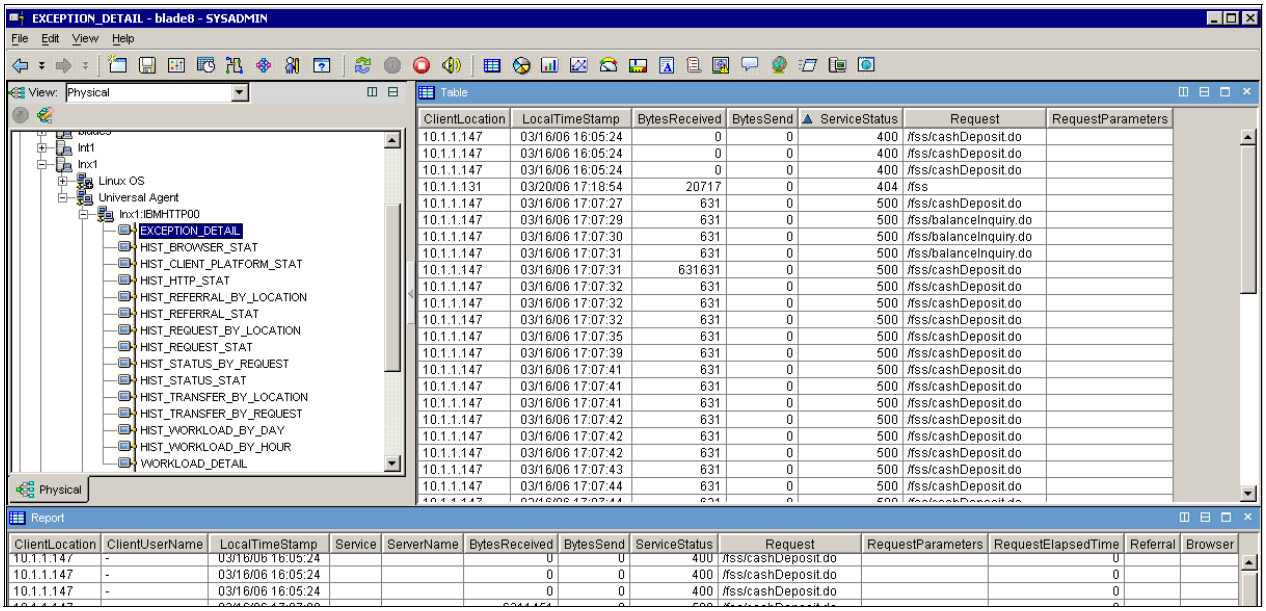

EXCEPTION_DETAIL - blade8 - SYSADMIN
File Edit View Help
View: Physical
Table
Report
Physical
Linux OS
Universal Agent
lnx1:IBMHTTP00
EXCEPTION_DETAIL
HIST_BROWSER_STAT
HIST_CLIENT_PLATFORM_STAT
HIST_HTTP_STAT
HIST_REFERRAL_BY_LOCATION
HIST_REFERRAL_STAT
HIST_REQUEST_BY_LOCATION
HIST_REQUEST_STAT
HIST_STATUS_BY_REQUEST
HIST_STATUS_STAT
HIST_TRANSFER_BY_LOCATION
HIST_TRANSFER_BY_REQUEST
HIST_WORKLOAD_BY_DAY
HIST_WORKLOAD_BY_HOUR
WORKLOAD_DETAIL
ClientLocation LocalTimeStamp BytesReceived BytesSend ServiceStatus Request RequestParameters
10.1.1.147 03/16/06 16:05:24 0 0 400 /fss/cashDeposit.do
10.1.1.147 03/16/06 16:05:24 0 0 400 /fss/cashDeposit.do
10.1.1.147 03/16/06 16:05:24 0 0 400 /fss/cashDeposit.do
10.1.1.131 03/20/06 17:18:54 20717 0 404 /fss
10.1.1.147 03/16/06 17:07:27 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:29 631 0 500 /fss/balanceInquiry.do
10.1.1.147 03/16/06 17:07:30 631 0 500 /fss/balanceInquiry.do
10.1.1.147 03/16/06 17:07:31 631 0 500 /fss/balanceInquiry.do
10.1.1.147 03/16/06 17:07:31 631631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:32 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:32 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:32 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:35 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:39 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:41 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:41 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:41 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:42 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:42 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:42 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:43 631 0 500 /fss/cashDeposit.do
10.1.1.147 03/16/06 17:07:44 631 0 500 /fss/cashDeposit.do
ClientLocation ClientUserName LocalTimeStamp Service ServerName BytesReceived BytesSend ServiceStatus Request RequestParameters RequestElapsedTime Referral Browser
10.1.1.147 - 03/16/06 16:05:24 0 0 400 /fss/cashDeposit.do 0
10.1.1.147 - 03/16/06 16:05:24 0 0 400 /fss/cashDeposit.do 0
10.1.1.147 - 03/16/06 16:05:24 0 0 400 /fss/cashDeposit.do 0

Figure 3-42 TEP: Connection exception on the IBM HTTP Server

- The attribute group WORKLOAD_DETAIL provides miscellaneous statistics about the amount of data transferred between the clients and the server during a period of 5 minutes. This short-term history is based on a feature of the file data provider, which helps summarize data for a specific period of time. For a better understanding of the data provided by this attribute group, briefly explain the content of the metafile used by the file data provider.

  Example 3-5 shows a partial listing of the IBMHTTP.mdl metafile, where the WORKLOAD_DETAIL attribute group is defined.

Example 3-5 Partial listing of the IBMHTTP.mdl meta file

```
//NAME Workload_Detail S 360
//INTERNAL INPUT IBMHTTPLogRecord
//SUMMARY  300 Force
//ATTRIBUTES
ClientLocation     D  252                              SEQ=1
ClientUserName     D  32                               SEQ=2
Authorized_User    K  32
-Date_Time         DL 20
LocalTimeStamp     (CandleTimeStamp = Date_Time)       SEQ=3
Time_Zone          K  5
-OperationName     D  32                                      DLM='" '
Request            D  252                              SEQ=9  DLM=' '
-ProtocolVersion   D  32                                      DLMSTREND='" '
ServiceStatus      C  99999999                         SEQ=8
BytesReceived      C  99999999                         SEQ=6  SKEY=SUM
* -ReferProtocol     D  4                                     DLM='":'
Referral           D  252                              SEQ=12 DLM='"'
Browser            D  252                              SEQ=13 DLM='"'
Service            D  32                               SEQ=4
```

```
ServerName         D  252                          SEQ=5
RequestParameters  D  252                          SEQ=10
BytesSend          C  99999999                     SEQ=7  SKEY=SUM
RequestElapsedTime C  99999999                     SEQ=11
Requests_PerSecond REAL   (_Occurrences / _Interval_Unit)
BytesReceived_PerSecond   (BytesReceived / _Interval_Unit)
BytesReceived_PerRequest  (BytesReceived / _Occurrences)
BytesSend_PerSecond   (BytesSend / _Interval_Unit)
BytesSend_PerRequest  (BytesSend / _Occurrences)
```

The //SUMMARY statement defines the summarization period of data input during monitoring. The summarization period cannot exceed 86400 seconds (1 day).

When using the //SUMMARY statement, the resulting output attribute group consists of the following:

- LocalTimeStamp

  This is a timestamp indicating the beginning of a summary interval

- Interval_Unit

  This is as defined in the //SUMMARY statement parameter (300 seconds)

- Occurrences (total count)

  This is the number of records summarized

- Key attributes

  All the key attributes. In this case, all the numeric attributes defined with SKEY=SUM (total result count for a summarizing interval).

To calculate the statistics, such as the number of bytes received per second, new attributes called derived attributes are created based on the default attribute group provided by the //SUMMARY statement. For example, the new attribute _Occurrences is derived from the default attribute, Occurrence. Moreover, derived attributes can be created based on the numeric attribute defined with SKEY=SUM (in this case, BytesSend and BytesReceived attributes).

The attribute groups HIST_WORKLOAD_BY_DAY and HIST_WORKLOAD_BY_HOUR are based on the concept. The difference is only the summarization period defined in the //SUMMARY statement set to 86400 seconds (1 day) and 3600 seconds (1 hour).

Figure 3-43 shows an example of data provided by the WORKLOAD_DETAIL attribute group. The default report has been customized a little for a better understanding of the published data.

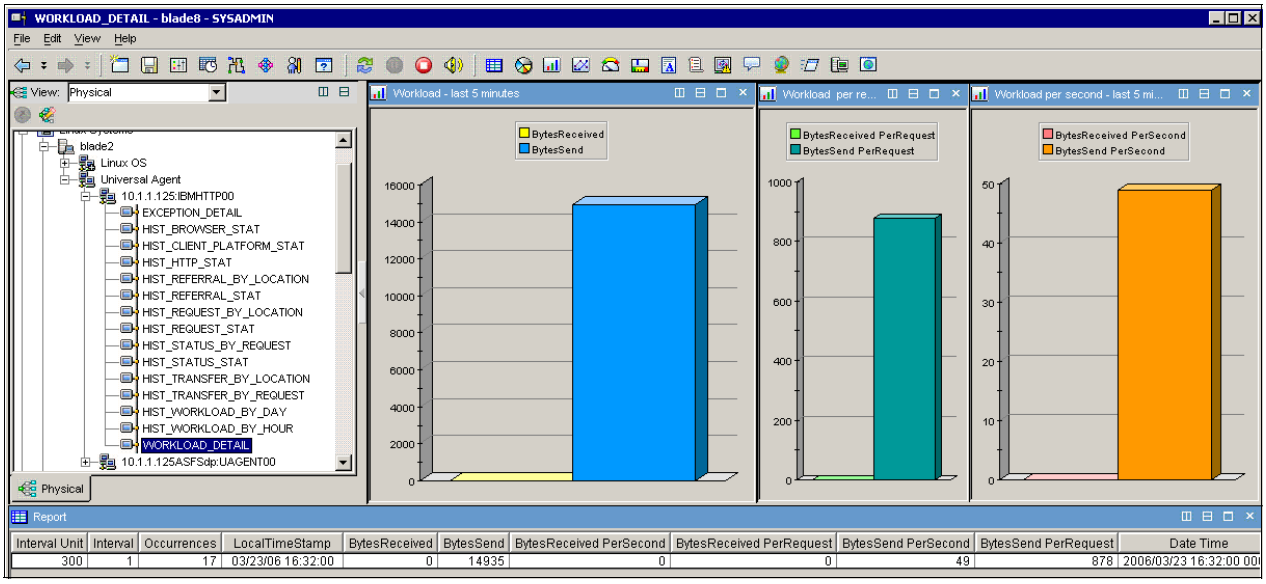


Figure 3-43 TEP: HTTP workload details

3.4.4 Simple network monitoring

The Simple Network Management Protocol (SNMP) data provider available in the Universal Agent provides information pertaining to the network environment, such as nodes availability, nodes description, and so on. The installation and settings are described in 2.2, “The distributed environment” on page 84. In our environment, monitoring is focused only on the production network (network 10.1.1.0 / 24).

The content of the attribute groups relates to the settings defined during the configuration of the SNMP data provider.

The managed system <hostname>:SNMP-MANAGER00 has two attribute groups containing data (the others are empty):

- The first group called ROUTER, contains a set of information pertaining to the routers’ availability. A router may have the following status:
  - Verify

    The SNMP Data Provider is in the process of verifying router status.
  - Online

    The router is active and operational.
  - Offline

    The router is not operational.
  - Passive

    The router is a daemon and is not actively participating in the network operation.

  When starting, the SNMP data provider reloads the already discovered routers from the KUMSROUT file.

Figure 3-44 shows the routers' status. Because the production network is isolated from the other networks, there is no active router. The router detected on the intranet network 9.121.131.0 is the one used as the default gateway on the server running the Universal Agent.

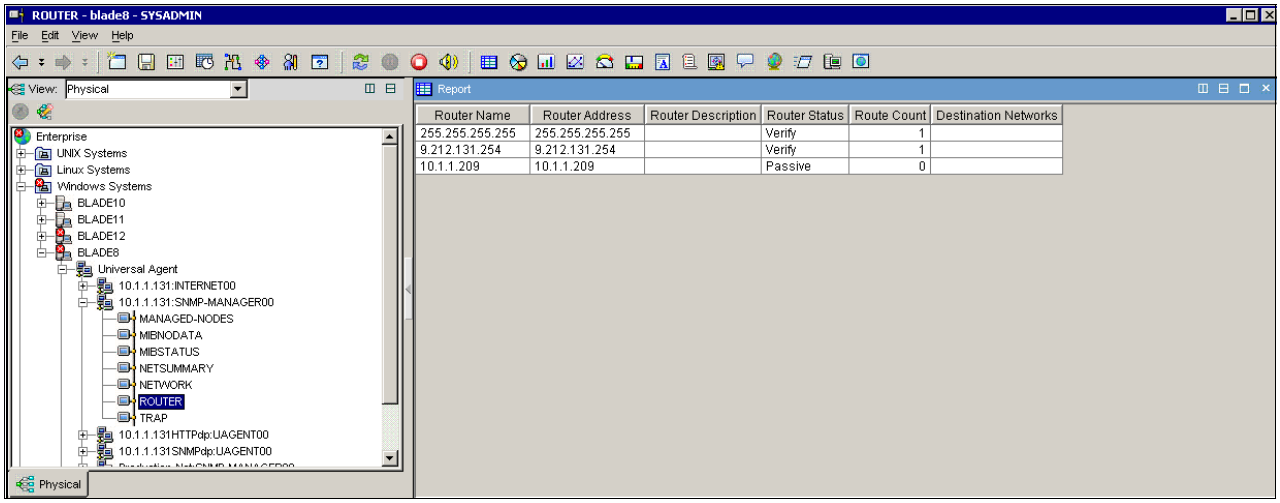


Figure 3-44 TEP: Router status

- The second attribute group called NETSUMMARY is useful to quickly determine which networks are monitored by the SNMP data provider and get simple information such as the number of nodes detected, and the response time statistics.

  When starting the SNMP, the data provider reloads the already discovered networks and their attributes from the KUMSNETS file.

  Figure 3-45 shows the network summary. Only the production network is managed by the SNMP data provider.

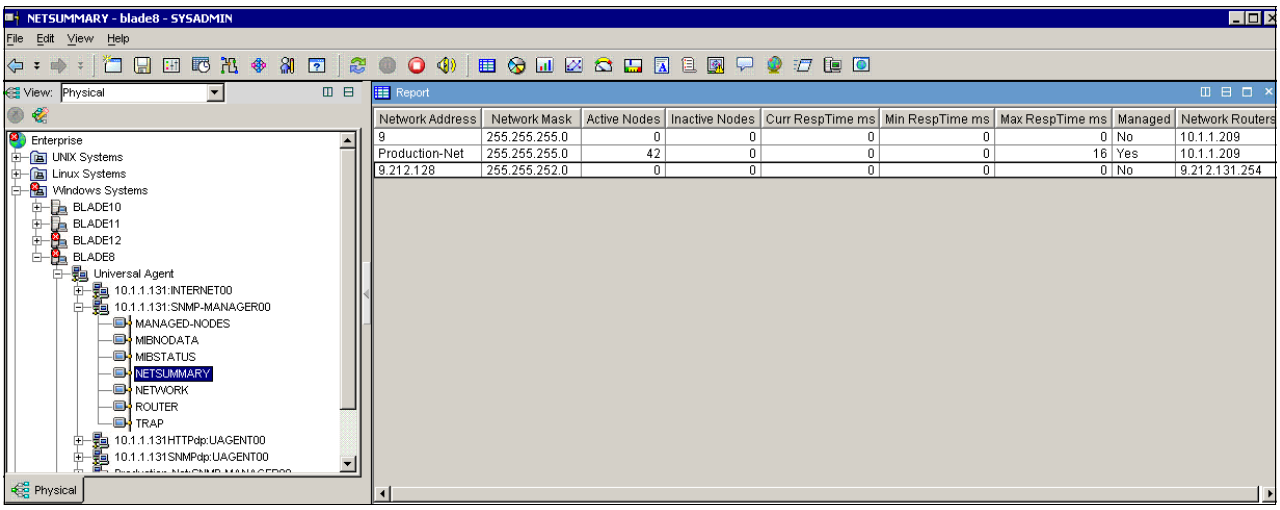

NETSUMMARY - blade8 - SYSADMIN
File Edit View Help
View: Physical
Report
Network Address | Network Mask | Active Nodes | Inactive Nodes | Curr RespTime ms | Min RespTime ms | Max RespTime ms | Managed | Network Routers
9 | 255.255.255.0 | 0 | 0 | 0 | 0 | 0 | No | 10.1.1.209
Production-Net | 255.255.255.0 | 42 | 0 | 0 | 0 | 16 | Yes | 10.1.1.209
9.212.128 | 255.255.252.0 | 0 | 0 | 0 | 0 | 0 | No | 9.212.131.254
Enterprise
UNIX Systems
Linux Systems
Windows Systems
BLADE10
BLADE11
BLADE12
BLADE8
Universal Agent
10.1.1.131:INTERNET00
10.1.1.131:SNMP-MANAGER00
MANAGED-NODES
MIBNODATA
MIBSTATUS
NETSUMMARY
NETWORK
ROUTER
TRAP
10.1.1.131HTTPdp:UAGENT00
10.1.1.131SNMPdp:UAGENT00
Physical

Figure 3-45 TEP: Network summary

The managed system Production-Net:SNMP-MANAGER00 contains an attribute group containing data (the others are empty). This managed system is created by the Universal Agent based on the information provided in the KUMSNAME file. This file is updated during the setup of the SNMP data provider. The attribute group contains useful information pertaining to the nodes available on the production network.

- Name attribute

  This contains the host name of the node as defined in the Domain Name System (DNS) or hosts file or the TCP/IP address only if the host name cannot be resolved (not defined in DNS or hosts file). However, the TCP/IP address is always displayed even if no node on the network used this address. In such a situation, the Status attribute is Unknown.

- Status attribute

  When the node is up, the status is Online. Otherwise, the status is always Unknown. There is no way of making a distinction between node down (meaning, it is now Offline, but was up in the past) and the nonexisting node.

- Type attribute

  If SNMP is enabled on the remote node, the SNMP data provider is able to determine the node type, router, bridge, and so on. The data provider must know the SNMP read-only community authorized by the remote node to access the Management Information Base (MIB). The default community is Public. However, this may be different based on the information provided in the following:

  – The KUMP_SNMP_NET_COMMUNITY environment variable to define the default SNMP community name

  – The KUMSCOMM file to define the SNMP community name for a particular network address

- Location attribute

  The information is retrieved from the sysLocation object instance of MIB-II of the remote node.

- Description attribute

  The information is retrieved from the sysDescr object instance of MIB-II of the remote node.

Figure 3-46 shows the network details of the production network.

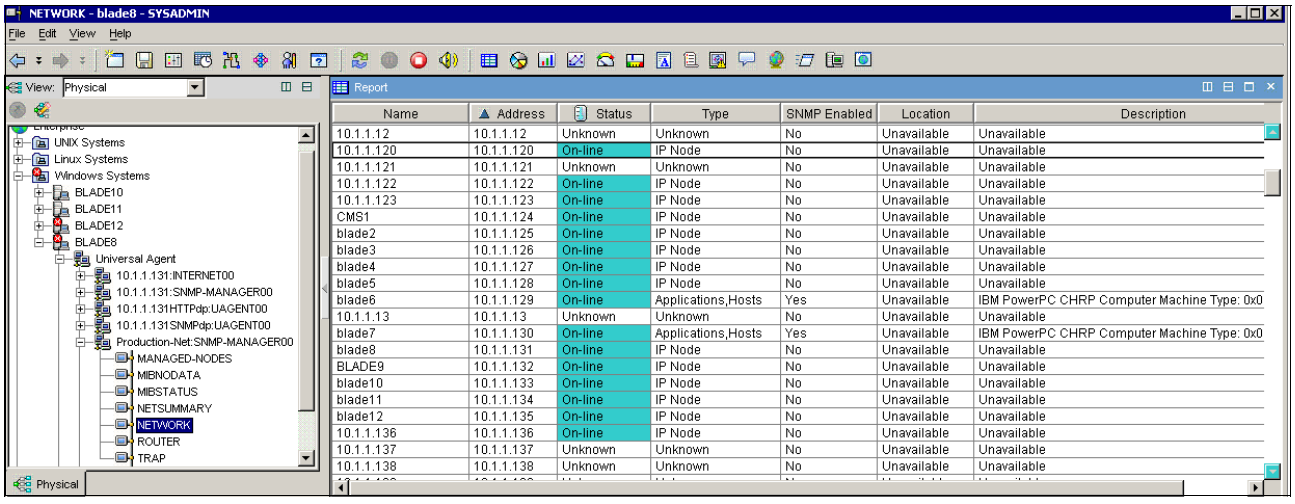

NETWORK - blade8 - SYSADMIN
File Edit View Help
View: Physical
Report
UNIX Systems
Linux Systems
Windows Systems
BLADE10
BLADE11
BLADE12
BLADE8
Universal Agent
10.1.1.131:INTERNET00
10.1.1.131:SNMP-MANAGER00
10.1.1.131HTTPdp:UAGENT00
10.1.1.131SNMPdp:UAGENT00
Production-Net:SNMP-MANAGER00
MANAGED-NODES
MIBNODATA
MIBSTATUS
NETSUMMARY
NETWORK
ROUTER
TRAP
Physical
Name | Address | Status | Type | SNMP Enabled | Location | Description
10.1.1.12 | 10.1.1.12 | Unknown | Unknown | No | Unavailable | Unavailable
10.1.1.120 | 10.1.1.120 | On-line | IP Node | No | Unavailable | Unavailable
10.1.1.121 | 10.1.1.121 | Unknown | Unknown | No | Unavailable | Unavailable
10.1.1.122 | 10.1.1.122 | On-line | IP Node | No | Unavailable | Unavailable
10.1.1.123 | 10.1.1.123 | On-line | IP Node | No | Unavailable | Unavailable
CMS1 | 10.1.1.124 | On-line | IP Node | No | Unavailable | Unavailable
blade2 | 10.1.1.125 | On-line | IP Node | No | Unavailable | Unavailable
blade3 | 10.1.1.126 | On-line | IP Node | No | Unavailable | Unavailable
blade4 | 10.1.1.127 | On-line | IP Node | No | Unavailable | Unavailable
blade5 | 10.1.1.128 | On-line | IP Node | No | Unavailable | Unavailable
blade6 | 10.1.1.129 | On-line | Applications,Hosts | Yes | Unavailable | IBM PowerPC CHRP Computer Machine Type: 0x0
10.1.1.13 | 10.1.1.13 | Unknown | Unknown | No | Unavailable | Unavailable
blade7 | 10.1.1.130 | On-line | Applications,Hosts | Yes | Unavailable | IBM PowerPC CHRP Computer Machine Type: 0x0
blade8 | 10.1.1.131 | On-line | IP Node | No | Unavailable | Unavailable
BLADE9 | 10.1.1.132 | On-line | IP Node | No | Unavailable | Unavailable
blade10 | 10.1.1.133 | On-line | IP Node | No | Unavailable | Unavailable
blade11 | 10.1.1.134 | On-line | IP Node | No | Unavailable | Unavailable
blade12 | 10.1.1.135 | On-line | IP Node | No | Unavailable | Unavailable
10.1.1.136 | 10.1.1.136 | On-line | IP Node | No | Unavailable | Unavailable
10.1.1.137 | 10.1.1.137 | Unknown | Unknown | No | Unavailable | Unavailable
10.1.1.138 | 10.1.1.138 | Unknown | Unknown | No | Unavailable | Unavailable

Figure 3-46 TEP: Production network details

In the managed system SB showcase, SNMP-MANAGER00 contains one attribute group containing data (the others are empty). This managed system is created by the Universal Agent based on the information provided in the kumlist_Smart Bank showcase.def file. This file contains the managed node lists (Hot Lists), and is created during the setup of the SNMP data provider. The attribute group contains useful information regarding the availability and response time of nodes participating in the Smart Bank environment.

- Name attribute

  Contains only the host name specified in the managed node lists (Hot Lists)

- Node Status attribute

  When the node replies to the ping, the status is Online. Otherwise, the status is Offline.

- Status Time Stamp attribute

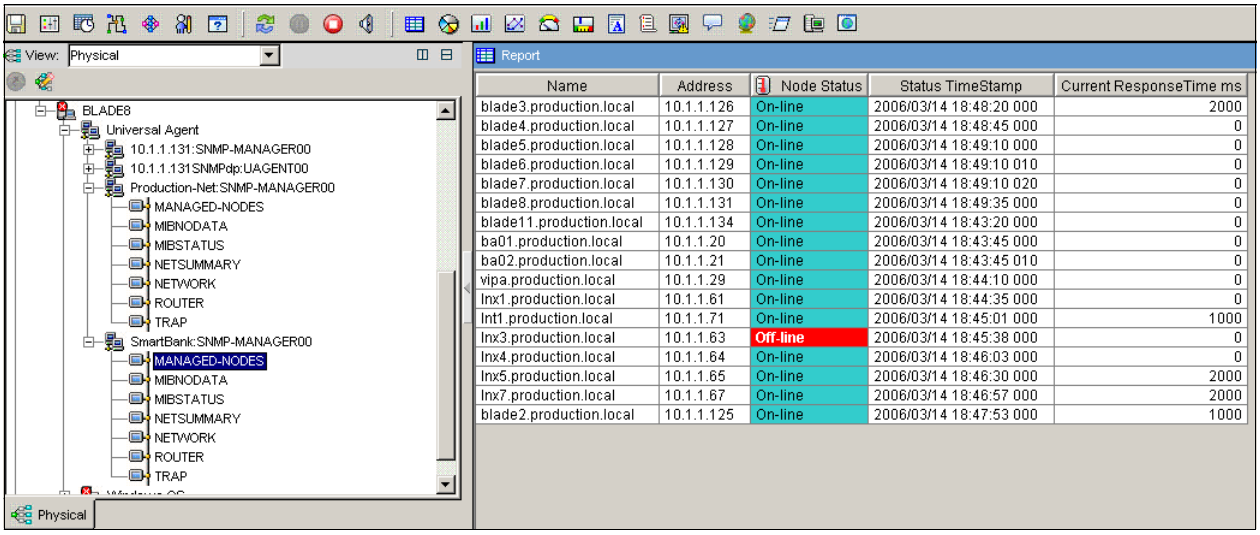
  Indicates when the ping is issued to check if the node is up.

> **Note:** According to *IBM Tivoli Monitoring Universal Agent User's Guide*, SC32-9459, a managed node list (Hot List) issues a ping at 30-second intervals to every device in the list. However, the Status Time Stamp field is updated every 5 minutes. No test is undertaken to determine if the ping is really issued every 30 seconds.

- Current Response Time attribute

  The number of milliseconds to receive a reply from the remote host to the ping request

Figure 3-47 shows the availability of the Smart Bank servers. The default report is customized for a better understanding of the published data.



Name	Address	Node Status	Status TimeStamp	Current ResponseTime ms
blade3.production.local	10.1.1.126	On-line	2006/03/14 18:48:20 000	2000
blade4.production.local	10.1.1.127	On-line	2006/03/14 18:48:45 000	0
blade5.production.local	10.1.1.128	On-line	2006/03/14 18:49:10 000	0
blade6.production.local	10.1.1.129	On-line	2006/03/14 18:49:10 010	0
blade7.production.local	10.1.1.130	On-line	2006/03/14 18:49:10 020	0
blade8.production.local	10.1.1.131	On-line	2006/03/14 18:49:35 000	0
blade11.production.local	10.1.1.134	On-line	2006/03/14 18:43:20 000	0
ba01.production.local	10.1.1.20	On-line	2006/03/14 18:43:45 000	0
ba02.production.local	10.1.1.21	On-line	2006/03/14 18:43:45 010	0
vipa.production.local	10.1.1.29	On-line	2006/03/14 18:44:10 000	0
lnx1.production.local	10.1.1.61	On-line	2006/03/14 18:44:35 000	0
lnt1.production.local	10.1.1.71	On-line	2006/03/14 18:45:01 000	1000
lnx3.production.local	10.1.1.63	Off-line	2006/03/14 18:45:38 000	0
lnx4.production.local	10.1.1.64	On-line	2006/03/14 18:46:03 000	0
lnx5.production.local	10.1.1.65	On-line	2006/03/14 18:46:30 000	2000
lnx7.production.local	10.1.1.67	On-line	2006/03/14 18:46:57 000	2000
blade2.production.local	10.1.1.125	On-line	2006/03/14 18:47:53 000	1000

Figure 3-47 TEP: Availability of the Smart Bank nodes

4

Customizing the Logical views

The objective of a Logical view is to select and reorganize parts of the physical data in order to visualize the status of the physical and logical elements involved in each of the applications.

This chapter discusses the following topics:

- 4.1, “Introducing the Logical view” on page 182 defines a Logical view
- 4.2, “Assigning managed systems to a Logical view” on page 189 describes how to set up managed systems to a Logical view
- 4.3, “Assigning situations to a Logical view” on page 193 describes how to assign a situation to a Logical view
- 4.4, “Defining the links” on page 201 discusses links
- 4.5, “Style in the Logical view” on page 241 discusses how to use the style features
- 4.6, “Logical view: Lessons learned” on page 252 summarizes the recommendations

© Copyright IBM Corp. 2006. All rights reserved.

4.1 Introducing the Logical view

When defining a Logical view, Tivoli Enterprise Portal (TEP) administrators can create their own tree. By default, this tree is completely disconnected from the physical tree shown in a Tivoli Enterprise Console. A part of the branch or all the branches of this new tree is then connected to data or alerts or both data and alerts that are gathered by the available Tivoli Enterprise Monitoring Agent. To do so, on each level of this tree, the administrator defines a workspace. The purpose of this workspace is to concentrate and highlight all the information that is required to supervise a specific collaboration between physical components.

Inside each workspace, the administrator defines views. Each view provides detailed information about the status of a specific physical element.

The final objective of a Logical view is to present relevant views in different points of the newly created tree. By extension, the entire process, the tree definition with the embedded levels and the workspaces set up with view definitions, is called the *definition of Logical view*.

4.1.1 Things to be considered before defining the Logical view

In order to maximize the use of Logical views, a few things must be considered before defining the views. The TEP user used to create the Logical views must have the relevant authority. In the Administer Users menu, this user must have the following permissions:

- User administration
  - Author mode eligible
  - Administration mode eligible
- Workspace administration
  - Workspace administration mode

    The Workspace administration mode authorizes this user to create TEP workspaces that can be shared by all the users
  - Workspace author mode

Before creating a new Logical view, the administrator must decide the architecture that is to be implemented, the information this Logical view shows, the tree structure, and so on.

4.1.2 Defining a Logical view

This section describes the step-by-step process involved in creating a Logical view.

Accessing the tree definition

1. Connect to the TEP. The default Physical view is displayed (Figure 4-1).

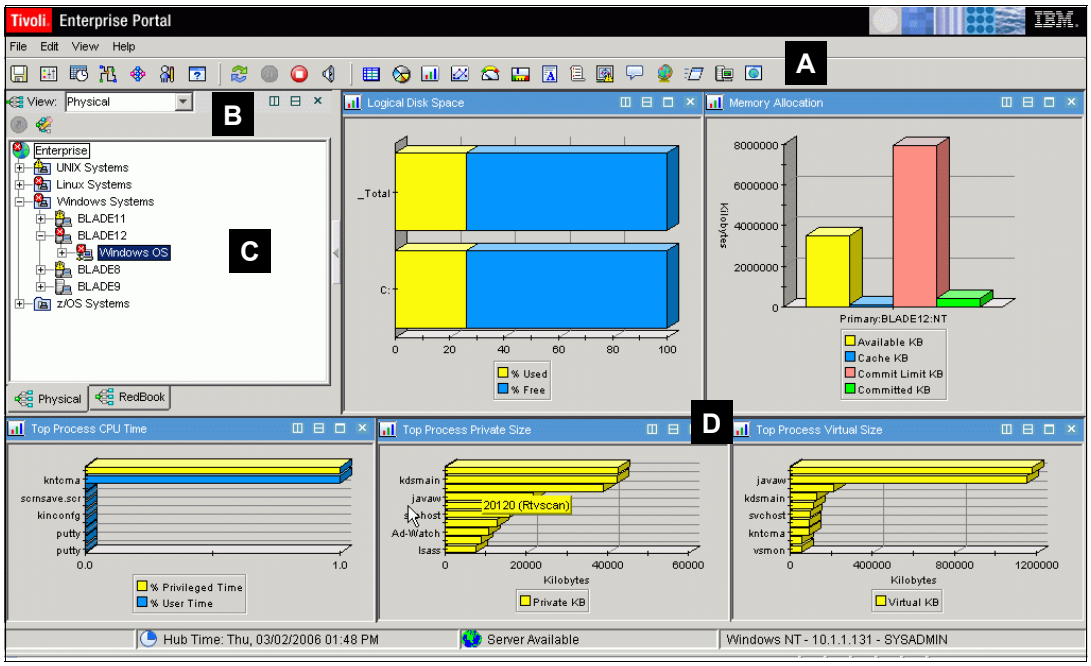


Figure 4-1 The default physical tree

There are four major areas in the TEP physical display:

- Area A provides access to the administrative functions and the tools used to build the views
- Area B shows the Navigator toolbar
- Area C is the physical tree
- Area D provides the views set up for the selected physical level in the tree

2. From the Navigator toolbar (B), access the Edit Navigator View shown in Figure 4-2. From the same point, select one of the Logical views you defined.

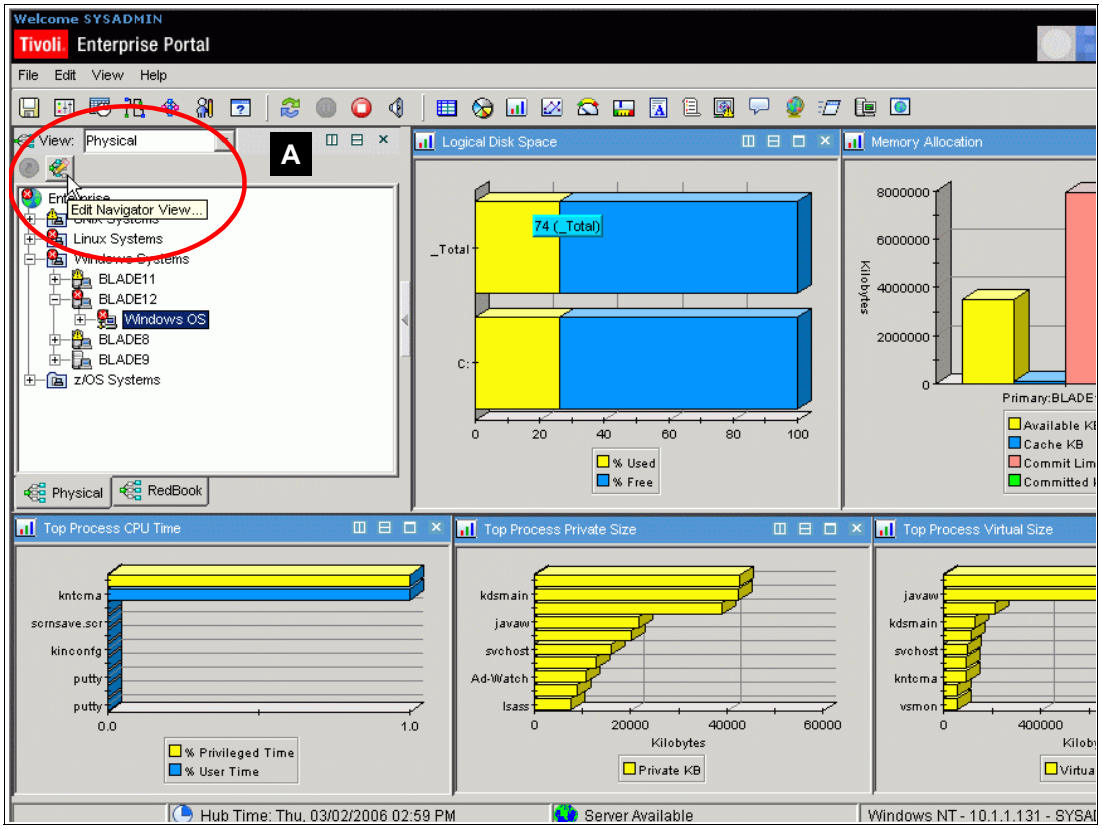


Figure 4-2 Accessing the Edit Navigator Tree function

3. Click the icon named Edit Navigator View in zone A. The TEP opens a new window with two frames.

Building a new Logical view

As required by the Smart Bank team, a Logical view that displays the status of all the channels used by the Smart Bank clients while dealing with the bank, is created.

1. Create a new Navigator View. Click the **Create New Navigator View** icon located in zone A, as shown in Figure 4-3.

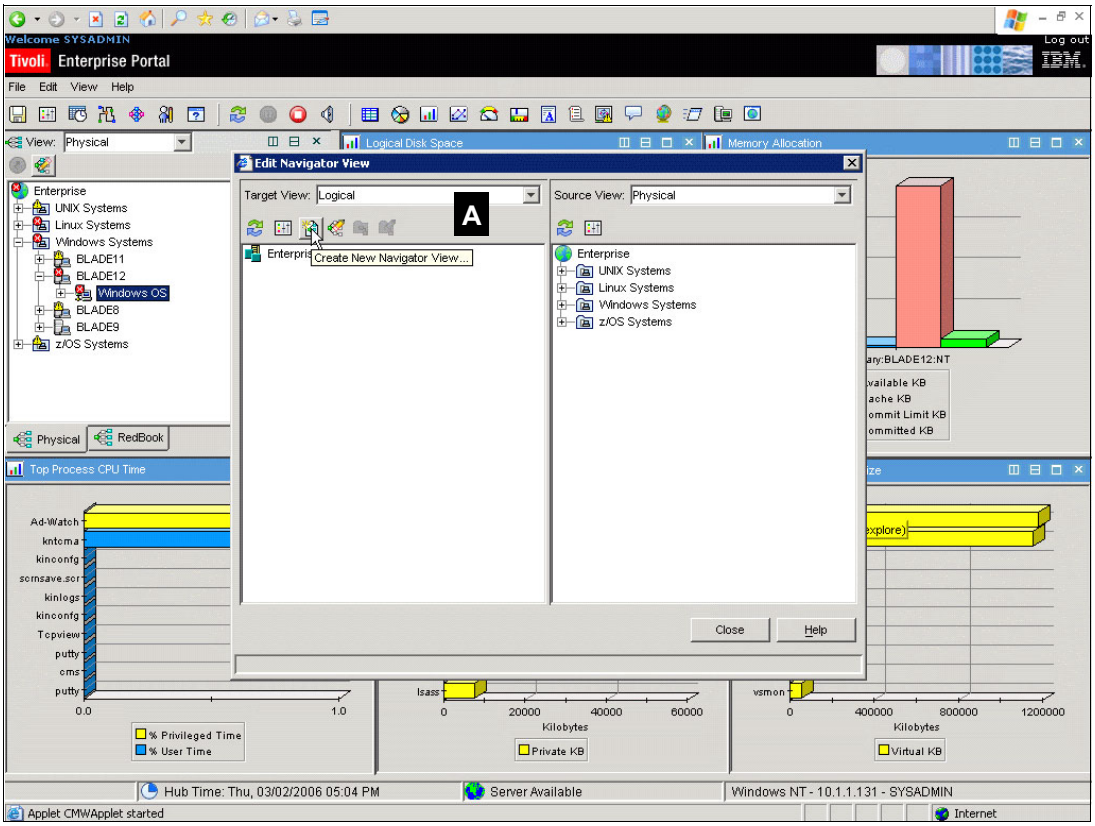


Figure 4-3 Starting the definition of a Logical view

2. A new window opens (Figure 4-4). Enter the name of the new Logical View, such as `Smart_Bank`, and a meaningful description. Click **OK.** The Logical view is now defined.

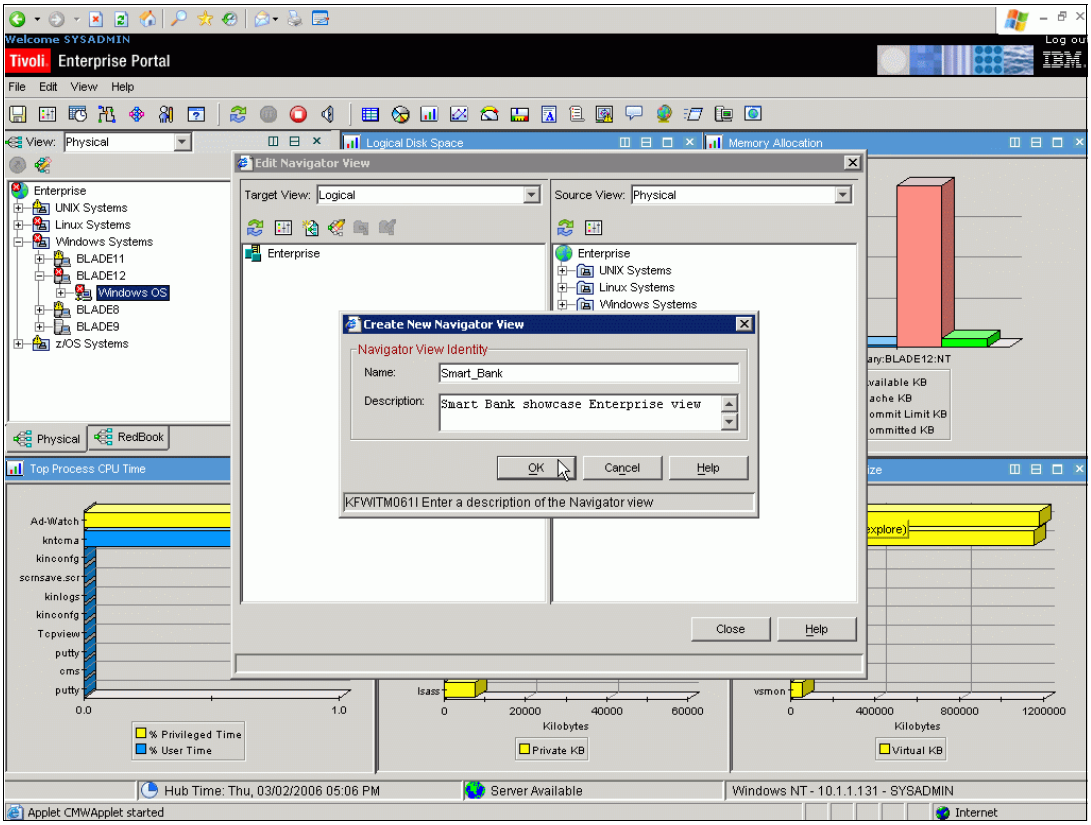


Figure 4-4 Defining the new Logical view

3. Define the embedded levels that must be a part of this Logical view by using the **Create Child Item** icon (zone A in Figure 4-5). A child is created under the current selected level (an outline is displayed around its name). It is possible to create a new child at any existing level of the actual Navigator view. Enter the name of the new level and a short description (zone B in Figure 4-5). The name given to the child item must be meaningful for all those who use this Logical view. After creating the new logical tree, click **Close**.

> **Note:** To build a Navigator view, provide a name first. Then, define children that are either attached directly to the name of the Navigator view or to an existing child. The *root* is the Navigator view, the *branches* are the children attached to the root, and the *leaves* are the children without attachments. By extension, a Navigator view that represents a Logical view is called a *logical tree*. A level in this tree is either the root, a branch, or a leaf.

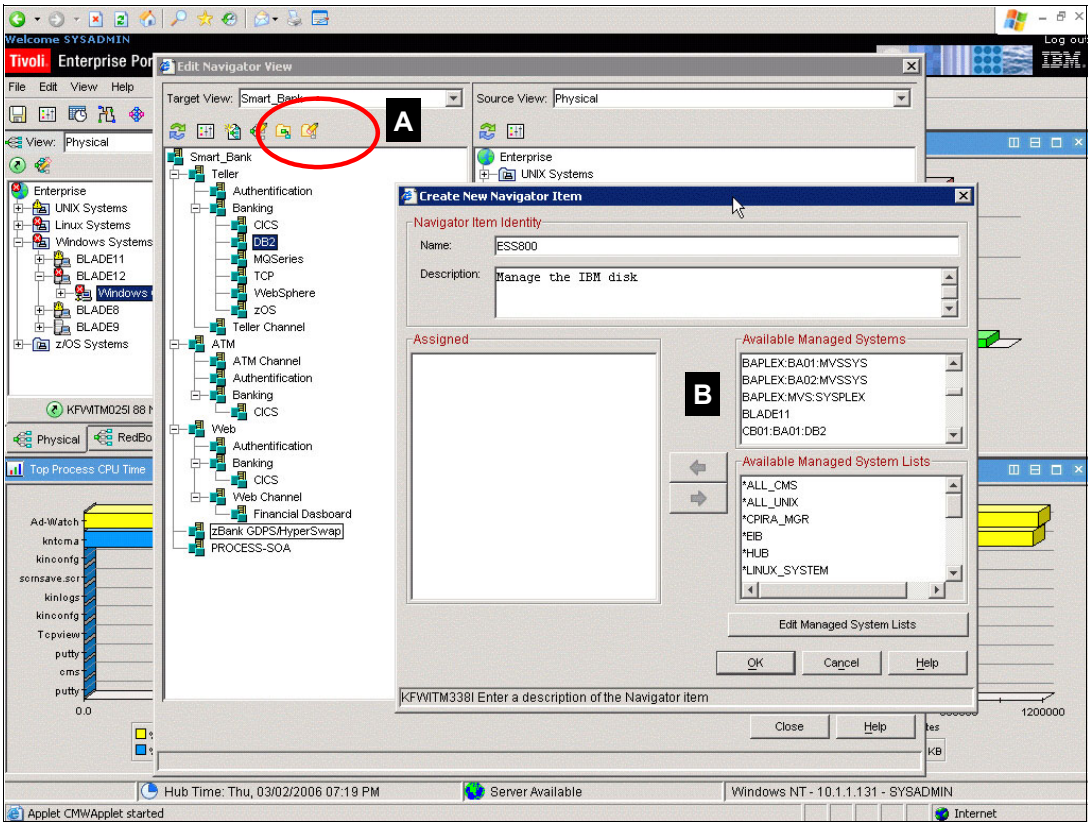


Figure 4-5 Creating a child in the Logical view

The Logical view is created, as shown in Figure 4-6.

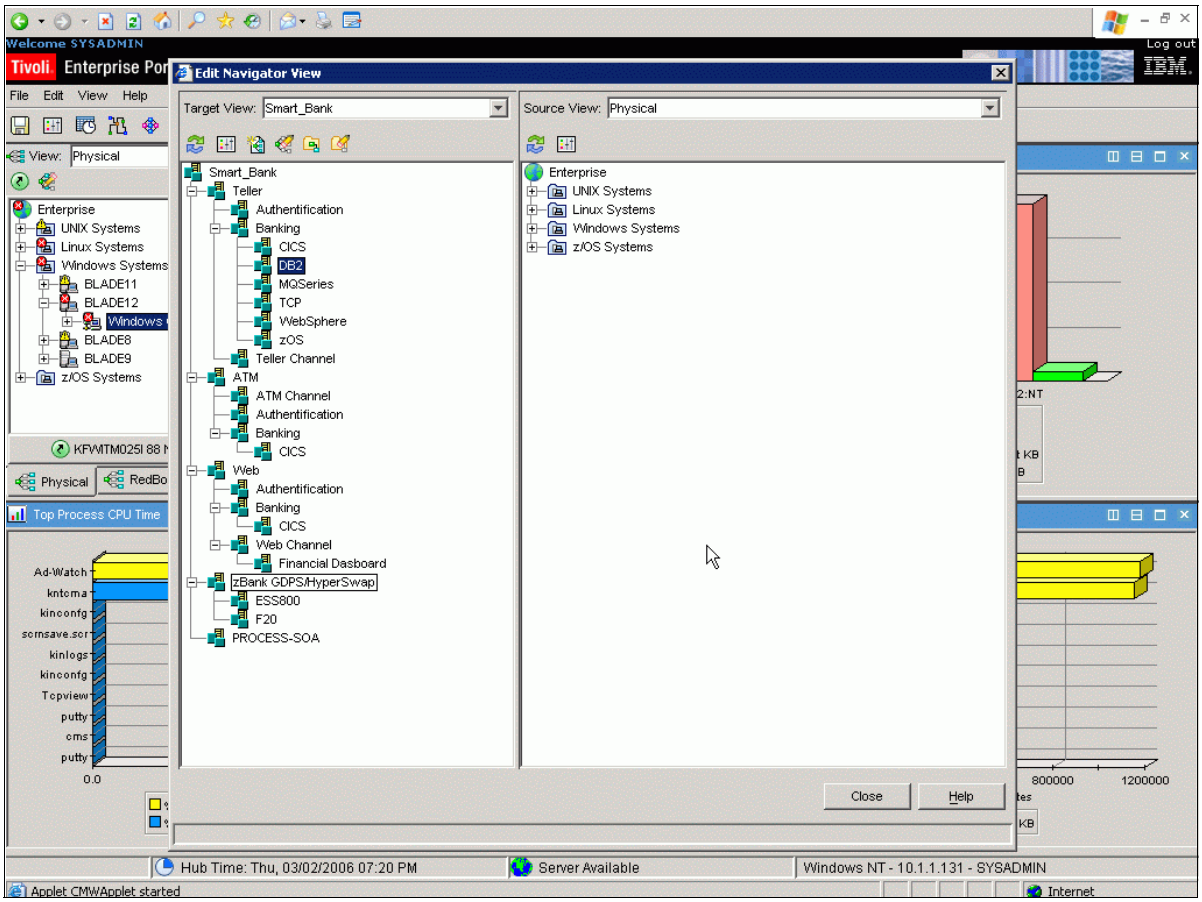


Figure 4-6 The completed Logical view

As required by the Smart Bank TEP administrator, the sublevels are added to some of the channels in order to provide more specific information at these levels. This is an implementation of a drill-down approach. When an alert is received, the user expands this corresponding level to analyze from which sublevel the incident comes from.

Using the existing physical tree or the logical tree

When using the Edit Navigator view, select the appropriate source view in the right side of the window (in the Physical view or in an existing Logical view). Develop that view up to your point of interest. Drag this point and drop it into the left side of the window in your Logical view at the level you have chosen. This process adds a child, with content equal to the tree selected in the source window. The target source in the right side of the window can be changed at any time to add new children.

This process is an easy way of reorganizing physical data because it adds parts of the existing trees (physical or logical), their existing setup (queries, views, and situations) attached quickly.

Recommendations

A Logical view is a tool that is used on a daily basis. Therefore, this Logical view must be meaningful and easy to use. In order to achieve this, follow the rules listed here:

- Use meaningful names for each part of the tree. Use standards.
- Keep the depth of the tree at three or four levels. This enables users to find their way much more easily.
- Define the views on a workspace that deals with different managed systems *only* when their purpose is to combine information in order to fully explain the state of what is monitored at a particular level of the tree.
- Use other product functions to simplify the usage, for example, you may have many regions that you want to monitor using a Logical view. To do so, you may develop the same leaf in many places of the Logical view. However, instead of doing this, define only one workspace and use it as a target of dynamic links. For more details, refer to 4.4, “Defining the links” on page 201.
- Build graphics using background maps or pictures. A picture can be a simple colored square that is built using a presentation tool and saved as a TEP-supported file.

Note: In order to use a newly defined Logical view, TEP users who are already connected must log in again.

4.2 Assigning managed systems to a Logical view

The purpose of a Logical view is to reorganize the data available through the Tivoli Enterprise Monitoring Agents. Following are the three ways of setting up managed systems to a level of a Logical view in order to be able to define the views at this level:

- Assigning one or more managed systems to a specific level, and then building views linked to these managed systems
- During the Logical view definition process, directly attach to the level of the physical tree or the part of an existing logical tree that already has managed systems assigned
- Directly build a view on that level. When assigning the required query, go to the Query Result Source index. Check the option **Let user assigned explicitly** and then select the managed system on which the query must run.

Note: You may combine the three methods. If you do not use them, the views cannot display the data from any agent and situations cannot be assigned.

4.2.1 Assigning a managed system to a specific level

When creating a Logical view or when using a Logical view, right-click the desired level and use the **Properties** option to assign the managed systems (Figure 4-7).

Note: You can assign a *Managed System List* to a level. When selecting the query for a view on a level, check the **Let user assign explicitly** option in the Query Results Source option. Manually select the managed system or systems on which this query will apply. Because this is a Logical view definition, by default, the TEP does not propagate the selected query to all the managed systems in the managed system list.

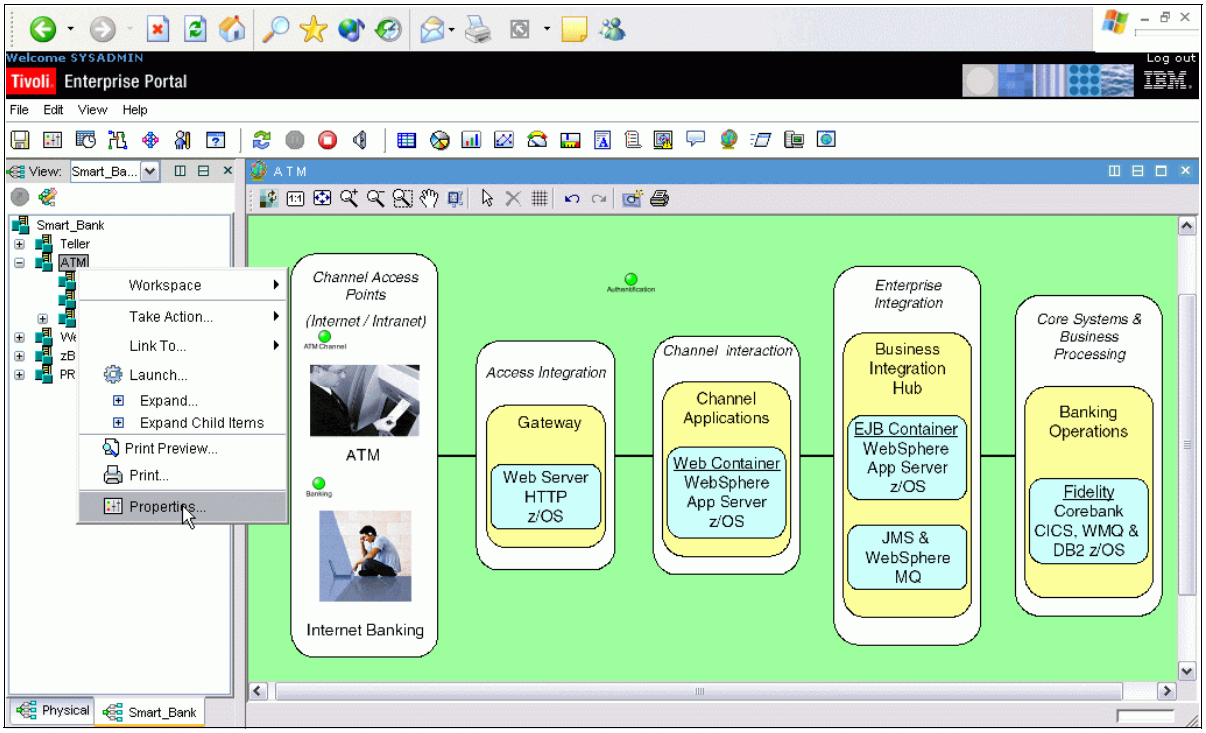


Figure 4-7 Properties access

Select the managed systems you want to apply from the menus in the right side (Figure 4-8).

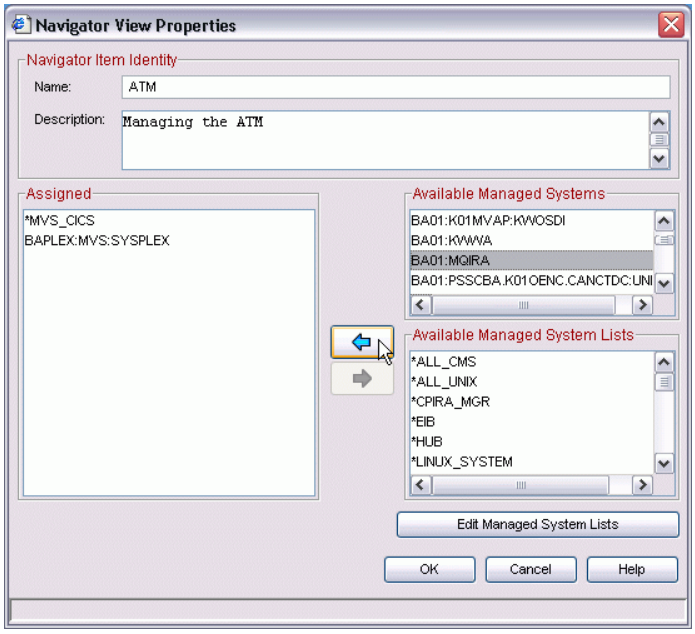


Figure 4-8 Assigning a managed system to a Logical view level

After this setup, in the workspace, build views that deal with one of these assigned managed systems. At this level, you can also manage situations that apply to all the selected managed systems.

4.2.2 Using managed system inheritance

During the Edit Navigator View process, you can copy a part of the existing Physical views or Logical views into your new Logical view. By doing this, you *inherit* the attached managed system from the existing setup, as also the queries, views, and situations.

In the Smart Bank showcase Logical view, at the ATM Banking Customer Information Control System-level (CICS-level), we copied the CICS regions directly from the physical tree. With this, we inherited the current assignment of the CICS regions (see the example for region CICSRA10 in Figure 4-9).

Note: A navigator tree works in an hierarchical manner. If a managed system is assigned to a specific level of the tree, all the other levels that are included up to the root, inherit from this managed system assignment.

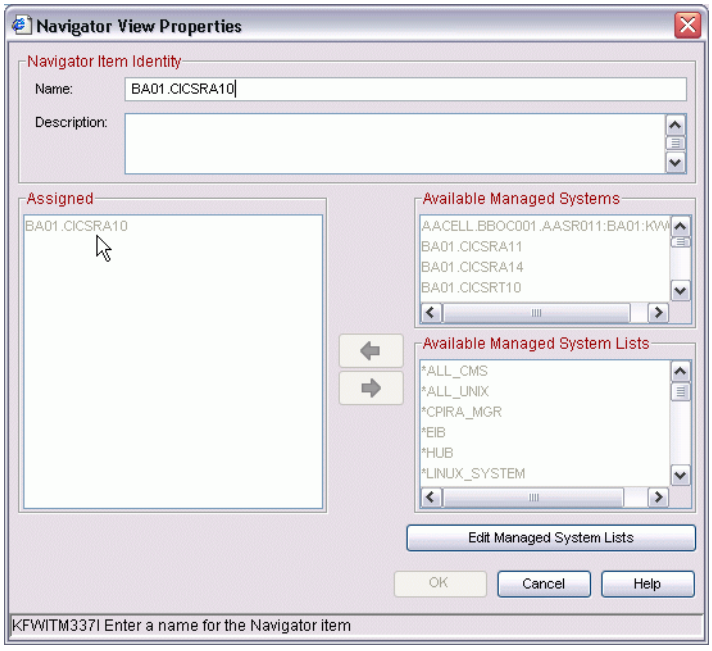


Figure 4-9 Inherited managed system

4.2.3 Adding views with explicit managed system

After adding a view, proceed as usual. To assign a query, select **Query Results Source**. Select the box against **Let user assign explicitly**. Select one or more Available Managed Systems from the list, as shown in Figure 4-10. Continue to define the view.

Attention: Although this option comes with the product, we do not recommend it. It links this view to specific managed systems. It is impossible to assign situations to this level of the Logical view when using only these modes of definition.

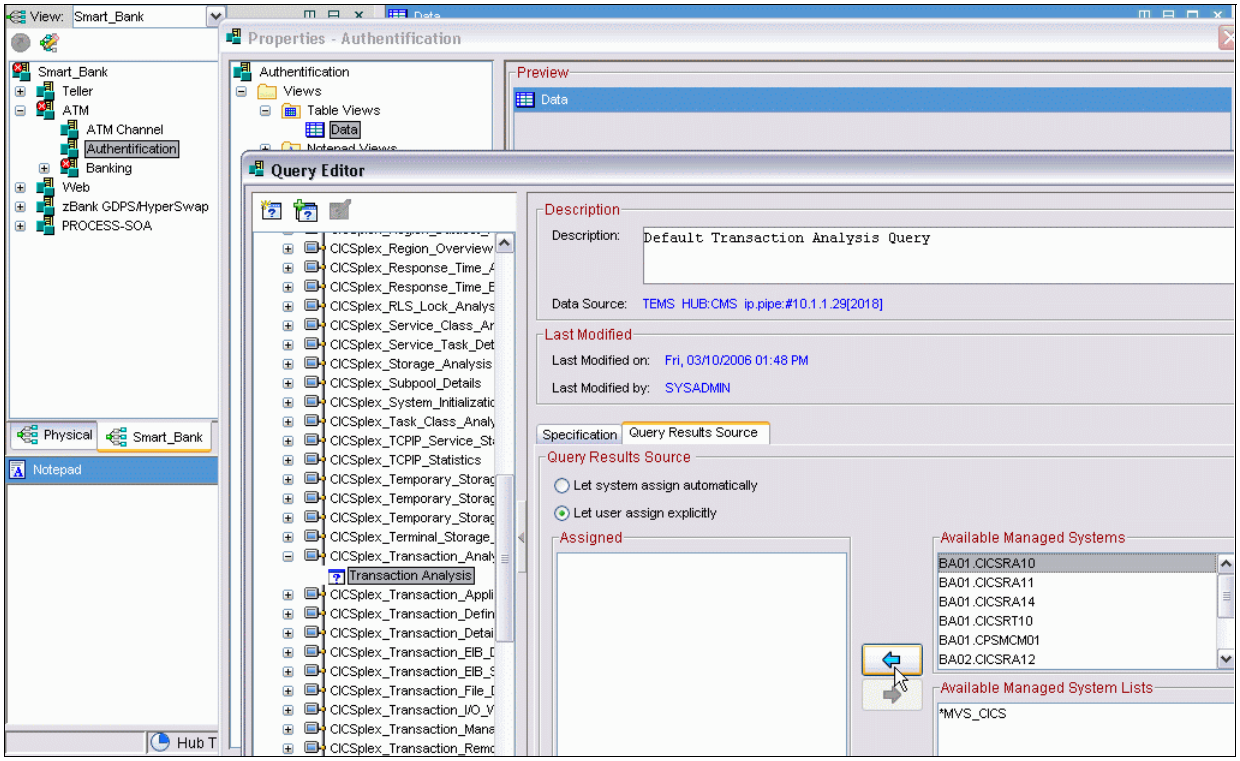


Figure 4-10 Adding views with the explicitly managed system

4.3 Assigning situations to a Logical view

Situations can be managed at each level of the new Logical view tree. When the Logical view is used, it is possible to perform the following tasks:

- Create a new situation
- Delete an existing situation
- Associate a situation
- Dissociate a situation

Note: If you build a new Logical view using a part of the other existing views, situations may already be assigned to the new Logical view.

You may create new situations and associate them to any level of the Logical view. There is no necessity to attach or create these situations under the physical tree.

Note: Because of these specific associations, a logical tree may display and react differently from a physical tree. The same rule applies if a part of the physical tree has been copied into the Logical view tree.

4.3.1 Creating and deleting a situation

To create or delete a situation, perform the following tasks:

1. At the chosen level, right-click and select the **Situations** option. If this option does not appear, it means that no managed system is attached to that level. Use the **Properties** option to do so.
2. The Situations for window opens (Figure 4-11). If some situations are already associated with this object, they are displayed.
   - Click the **Set Situation Filtering** icon located in zone A.
   - Select the **Eligible for Association** box to display all the situations that can be associated.
   - Click **OK**.

   The TEP sends back all the situations assigned to that level and relate to the managed system.

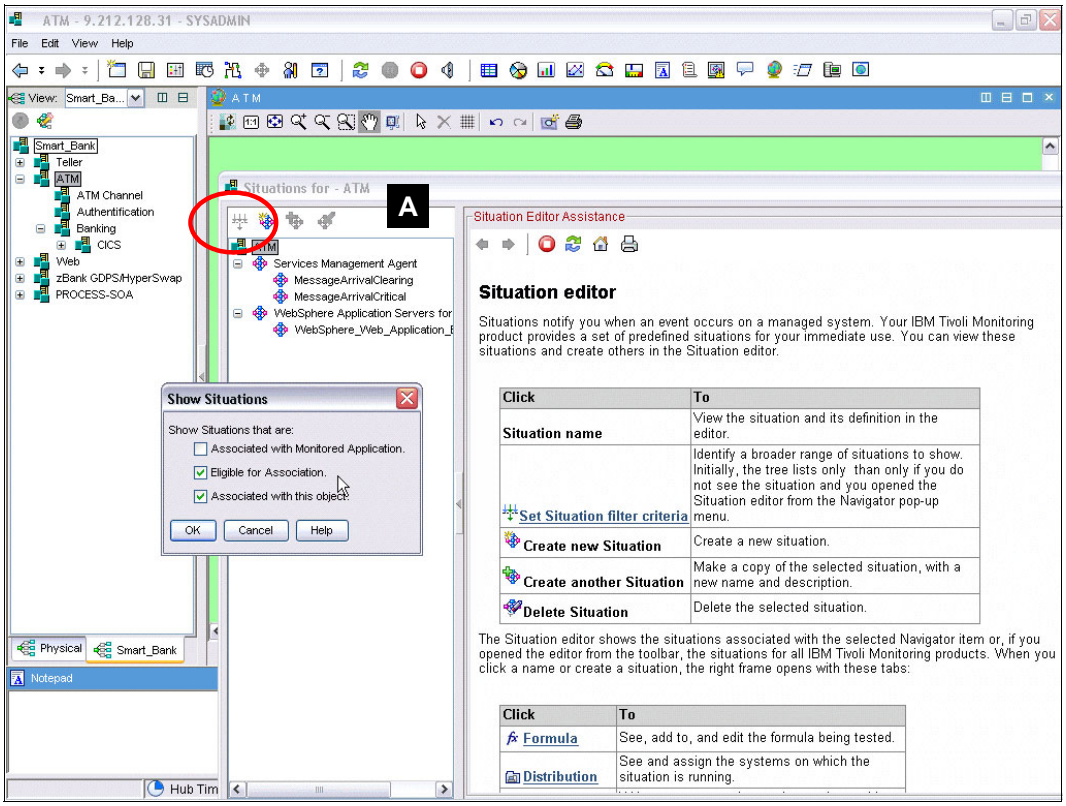

ATM - 9.212.128.31 - SYSADMIN
File Edit View Help
Smart_Bank
Teller
ATM
ATM Channel
Authentification
Banking
CICS
Web
zBank GDPS/HyperSwap
PROCESS-SOA
Situations for - ATM
A
Services Management Agent
MessageArrivalClearing
MessageArrivalCritical
WebSphere Application Servers for
Show Situations
Show Situations that are:
Associated with Monitored Application.
Eligible for Association.
Associated with this object.
OK
Cancel
Help
Physical
Smart_Bank
Notepad
Situation Editor Assistance
Situation editor
Situations notify you when an event occurs on a managed system. Your IBM Tivoli Monitoring product provides a set of predefined situations for your immediate use. You can view these situations and create others in the Situation editor.
Click
To
Situation name
View the situation and its definition in the editor.
Set Situation filter criteria
Identify a broader range of situations to show. Initially, the tree lists only than only if you do not see the situation and you opened the Situation editor from the Navigator pop-up menu.
Create new Situation
Create a new situation.
Create another Situation
Make a copy of the selected situation, with a new name and description.
Delete Situation
Delete the selected situation.
The Situation editor shows the situations associated with the selected Navigator item or, if you opened the editor from the toolbar, the situations for all IBM Tivoli Monitoring products. When you click a name or create a situation, the right frame opens with these tabs:
Click
To
Formula
See, add to, and edit the formula being tested.
Distribution
See and assign the systems on which the situation is running.

Figure 4-11 Eligible situations for association

3. You may then create a new situation, create another situation if an existing situation is selected, or delete a selected situation.

Note: In a Logical view, the management of situations (creating, deleting, and modifying) is performed the same way as the management of situations in a Physical view.

4.3.2 Associating and dissociating a situation

To associate or dissociate a situation, perform the following tasks:

1. As specified earlier, open the **Situations for** window. All the situations defined for the managed system associated with that level appear. Right-click the situation you want to associate to.

Attention: In the Situations for window, all the *defined* situations are displayed. All of them are *not* set up to run at startup or distributed the way you want them. We recommend that you double-check a situation's state before creating the association.

2. A new menu is displayed (Figure 4-12). Click the **Associate** option. (To terminate the association, click **OK** when the button is displayed.)

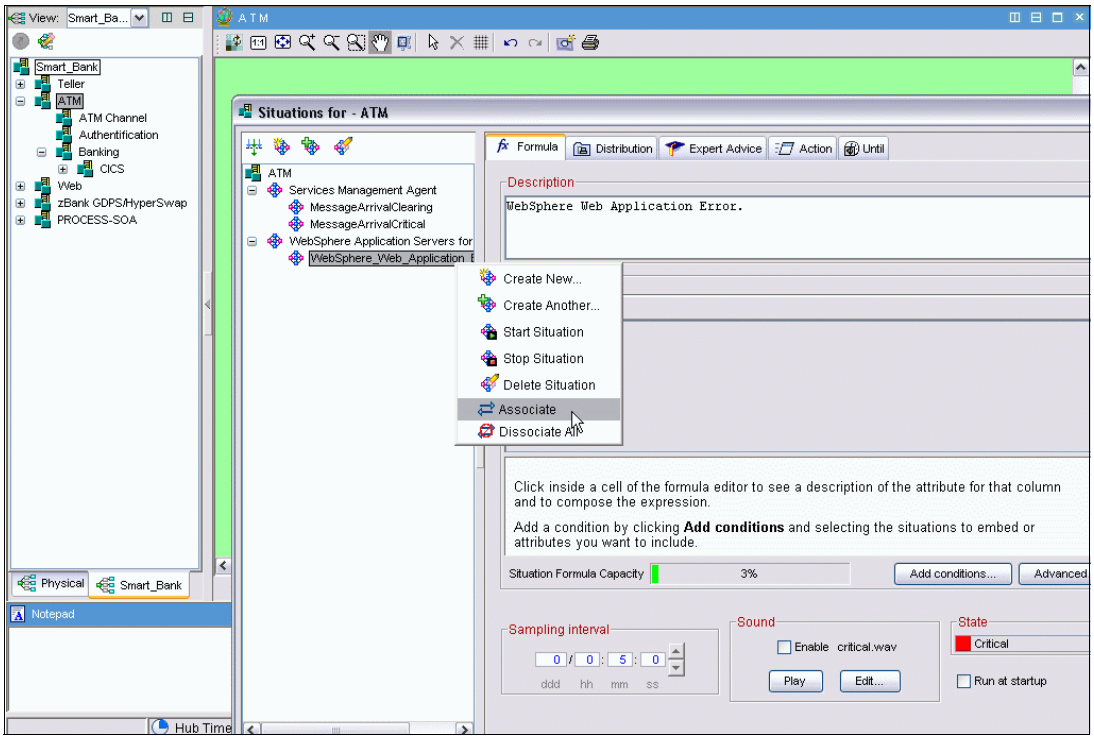


Figure 4-12 Associating a situation to a Logical view

This situation is now associated with the selected level of the logical tree. If this situation becomes true, an alert opens up at this logical tree level.

Dissociating a situation works in the reverse order. Select the **Dissociate** option to suppress the association at this level.

4.3.3 Advice in a situation

Advice relating to a situation provides explanation about what is happening and what can be done to solve a problem. This is especially interesting in a Logical view. There is more than one way to write a situation.

Advice using the ADVICE() function

The ADVICE function can be used in the Expert Advice definition part of a situation. It is called ADVICE("*kxx*:"+$ISITSTSH.SITNAME$). $ISITSTSH.SITNAME$ is a TEP variable. It contains the name of the situation. Replace *kxx* with the corresponding product code. At the time of use, this function replaces *kxx:* with the full path name. If you want to, for example, use the ADVICE function for a Linux situation on a TEP running on Windows, the path is T:\IBM\ITM\CNB\classes\candle\klz\resources\advice\en_us. The target file is in an HTML format and its name is identical to the situation name.

Notes: These files are implemented during installation. We recommend that you do *not* add new files in these directories in order to avoid future migration problems.

If you change the TEP language using the Language Support CD, change the subdirectory en_US to one corresponding to the language you have installed.

Advice as a user Web page

Because the objective of a logical tree is to present data in an efficient manner, it is a good idea to use the Web page approach directly in the ADVICE part of situation. However, using it disables the ability to support multiple languages.

You can generate the content of a Web page formatted advice by using any tool that is able to build a simple Web page. When editing a situation, copy the HTML content into the Expert Advice part of the situation directly. You can also add variables and set up their values using the TEP event attribute variables as shown in Example 4-1.

Example 4-1 Web page in a situation advice

```
name = $EVENT:ATTRIBUTE.ISITSTSH.SITNAME$;
node = $NODE$;
"<BODY BGCOLOR='##FEDCEF'><div align=center><h3><i>This is expert advice for </i>
</h3><h2>"+name+"</h2> on "+node+ "<p>Please notify your system administrator that
we have a <b>"+name+"</b> alert. <p>To learn more, click here: <a
href=http://www-306.ibm.com/software/sw-atoz/indexT.html> IBM Tivoli Products
</a></div>"
```

Copy this data into the advice setup of a situation, as shown in Figure 4-13.

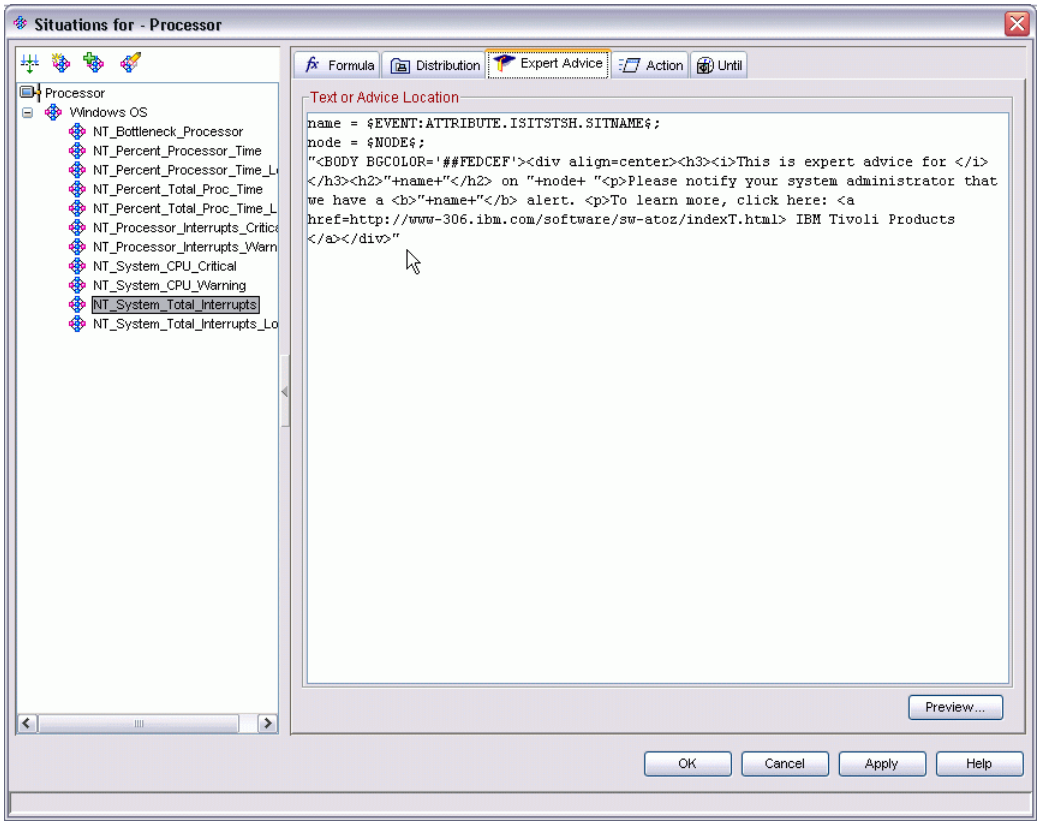


Figure 4-13 Web page advice in a situation

When used, a simple Web page is displayed, as shown in Figure 4-14.

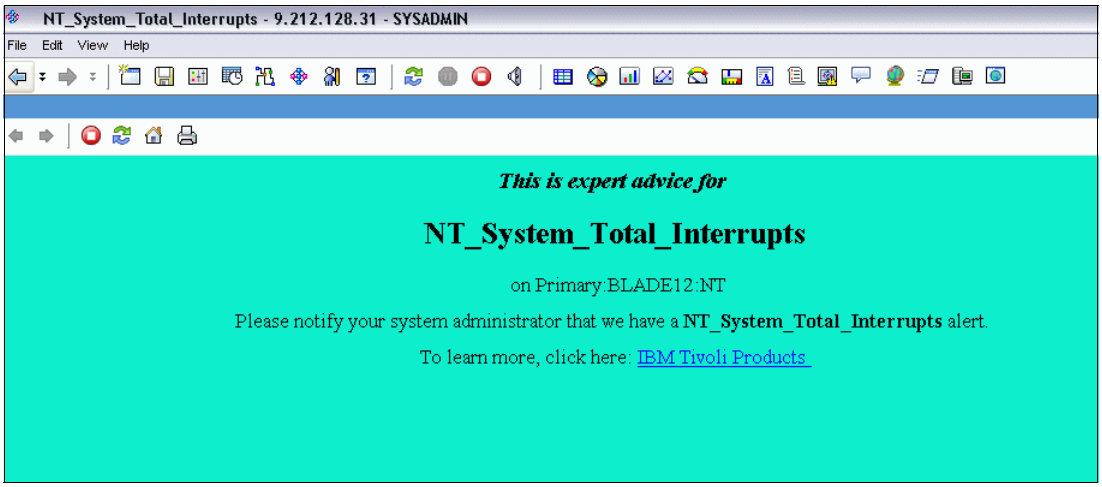

NT_System_Total_Interrupts - 9.212.128.31 - SYSADMIN
File Edit View Help
This is expert advice for
NT_System_Total_Interrupts
on Primary:BLADE12:NT
Please notify your system administrator that we have a NT_System_Total_Interrupts alert.
To learn more, click here: IBM Tivoli Products

Figure 4-14 Situation advice in a Web page format

The event attribute variables listed in Table 4-1 are available in a Web page advice.

Table 4-1 Attributes available in a Web page advice

Attribute	Variable
Situation name	$EVENT:ATTRIBUTE.ISITSTSH.SITNAMES$
Monitoring server name	$EVENT:ATTRIBUTE.ISITSTSH.NODE$
Managed system name	$EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$
Display item (if set)	$EVENT:ATTRIBUTE.ISITSTSH.ATOMIZE$
Global timestamp	$EVENT:ATTRIBUTE.ISITSTSH.GLBTMSTMP$
Local timestamp	$EVENT:ATTRIBUTE.ISITSTSH.LCLTMSTMP$
Status (Y/N)	$EVENT:ATTRIBUTE.ISITSTSH.DELTASTAT$

Tip: To find out the attribute names that can be used as variables, edit the situation. Select the **Show formula** option and check the box against **Show detailed formula**. The names that are now displayed are a concatenation of the group name, a period, and the attribute name. You can use it as a variable by surrounding it with two dollar ($$) signs.

Advice that prompts a user

The purpose of a Logical view is to display business information when managing alerts. The advice function prompts the users to focus on their demands and send back relevant data. Example 4-2 shows how to use the INPUT function.

Example 4-2 INPUT function in a situation advice

```
search = INPUT("What do you want to reasearch for?", "enter");"<BODY
BGCOLOR='#e6e6fa'>Searching database for <b>"+search+"</b>... <META
HTTP-EQUIV='refresh' CONTENT='1;
URL=http://www.mechanicalworks.com/seeq/int_results.jsp?portal_id=251&domain=mecha
nicalworks.com&keyword="+search+"'>"
```

1. Copy this into the Advice setup of a situation, as shown in Figure 4-15.

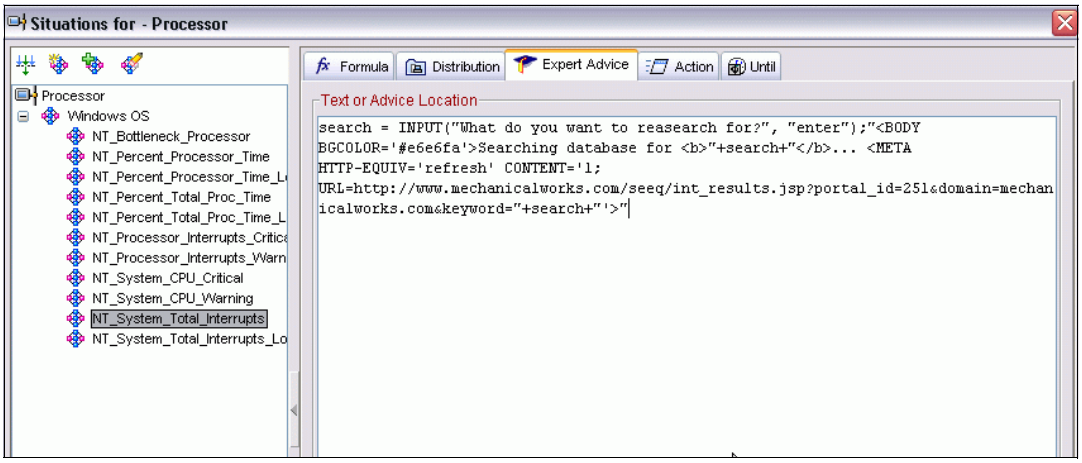


Figure 4-15 Advice in a situation using the INPUT function

2. When the situation becomes true and you select it to obtain details, you are prompted, as shown in Figure 4-16.

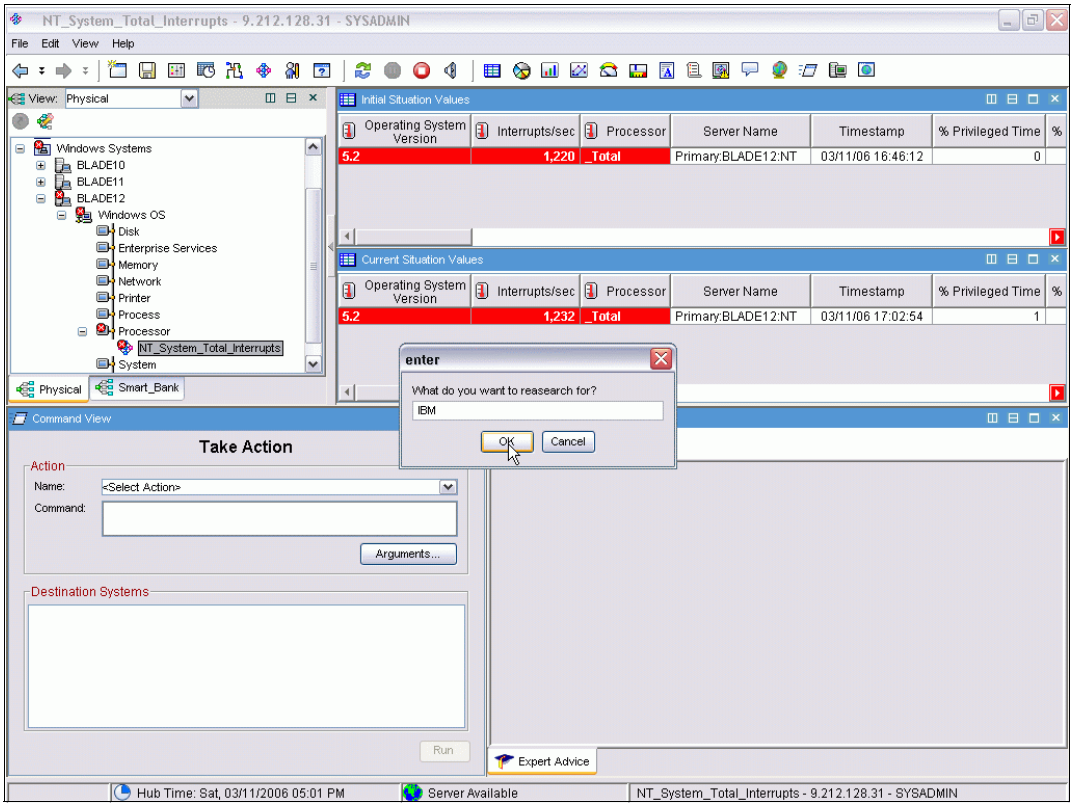


Figure 4-16 Prompt from the INPUT function in a situation advice

3. After filling the form, click **OK**. The next stage of the situation advice that runs at this juncture is shown in Figure 4-17.

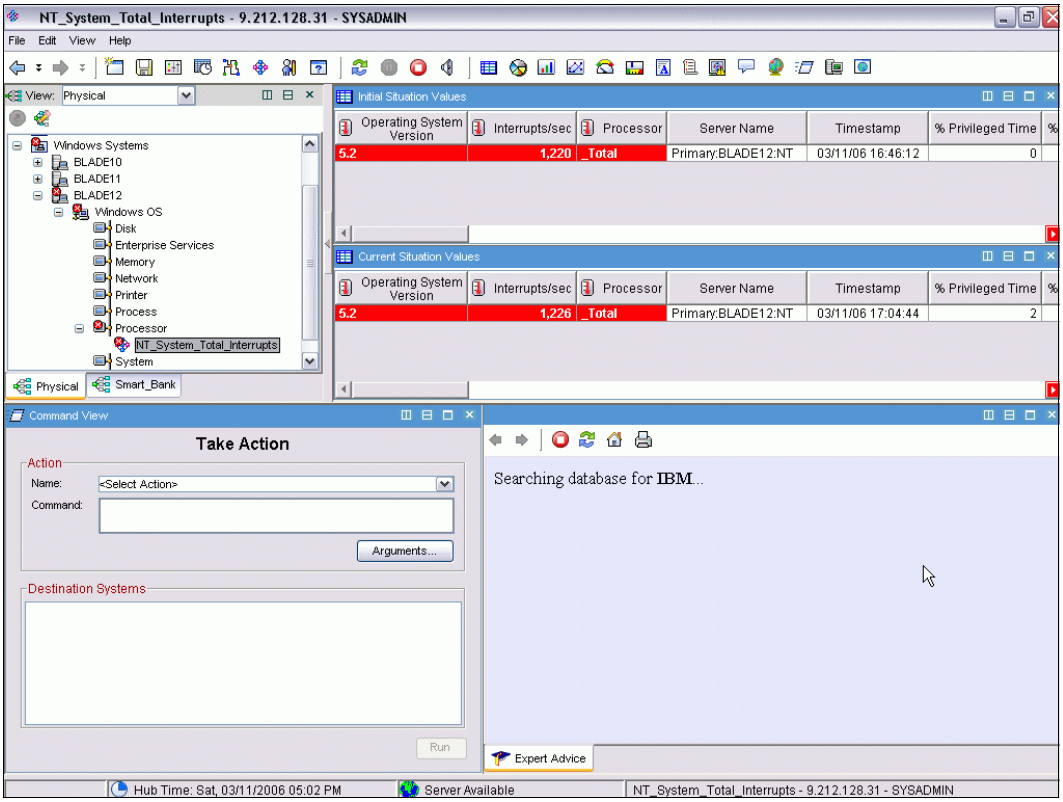


Figure 4-17 The next stage of situation advice

When the data comes back from the search URL, the data is displayed in the advice window (Figure 4-18).

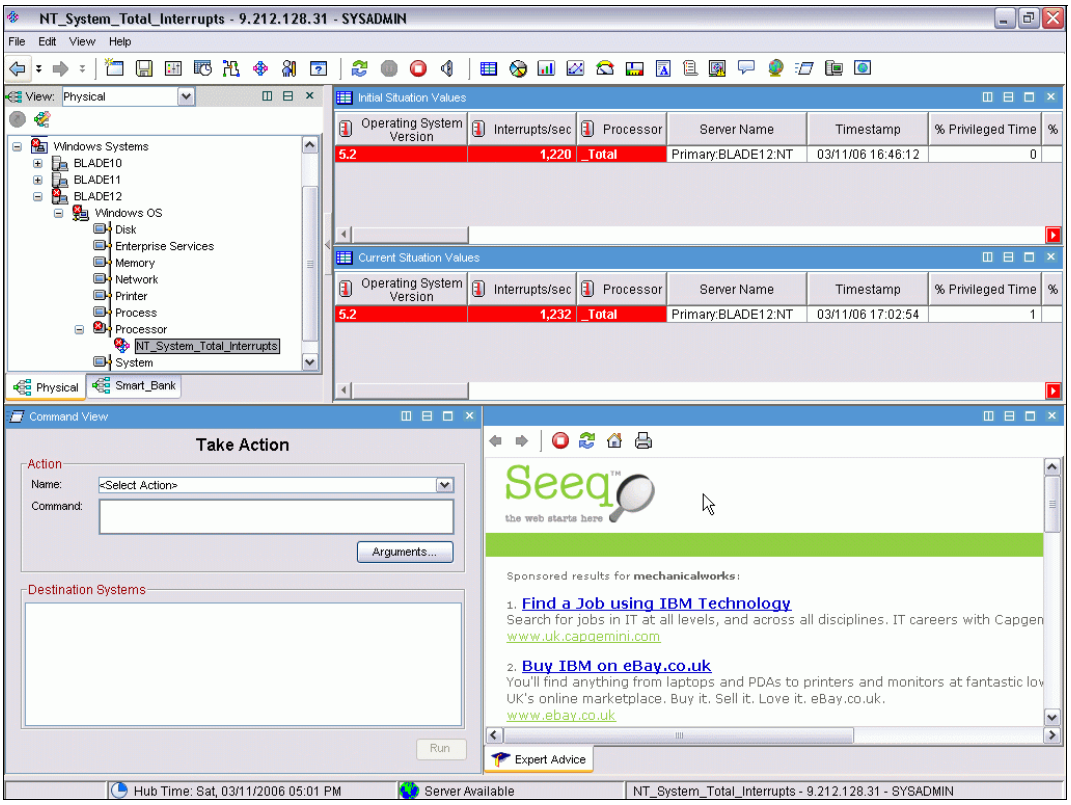


Figure 4-18 Result from the prompt in a situation advice

4.4 Defining the links

A Logical view displays data in an organized structure. Define *links* to navigate easily from one point of the structure to another point. This section explains the different type of links and how to define them.

4.4.1 Introducing the links

A link is an internal TEP definition. It is used by TEP users to easily navigate in the TEP organization. Links help you perform the following tasks:

- Jump from one view in a workspace to another one
- Drill down from one point to another point in order to obtain detailed information
- Keep track of the context in order to dynamically filter the target workspace that is displayed.

Following are the different link types:

- Absolute

  Jump from one source view to another view in a workspace

- Relative

  Jump from one source view to another view chosen in a list

- Advanced

  Jump from the source view and take out data in order to dynamically filter the target view

> **Note:** Before Fix Pack 2 (FP2), what is now called *Advanced* was called *Dynamic*. Dynamic Workspace Linking involves the ability to link from one product to another product and provide enough information in order to allow the proper target node to be specified. At the time of writing this book, the Smart Bank case did not implement the FP2. Therefore, the book sometimes refers to the Dynamic link, although the Advanced link is more appropriate. The Dynamic Workspace Linking, as provided by FP2, is *not* described in this book.

You can define a link in a Physical view or in a Logical view. Following are the types of view on which you can define a link:

- A level in a physical or a logical tree
- An icon in a graphic
- A row in a table
- A bar in a bar chart
- A slice in a pie chart
- A point in a plot chart

> **Note:** A view is always a part of a workspace. Thus, by using a link, you jump from one workspace to another workspace.

A link is accessible in the place it is defined, that is, from the selected view included in that workspace. When one or more links are defined on a table row or a graphic view icon, a *link anchor* is visible. For links defined on other elements, right-click the link and then select the link name you want to use.

To define a link, you require a source workspace and a target workspace. If you are linking from a data element such as a row or a bar, the source workspace must have the data being displayed before the link is defined. Some workspaces have data only when something exceptional happens. Therefore, in order to define a link, the target workspace *must* be built earlier.

After defining the tree structure of a Logical view, the administrator must also decide which link definitions are useful, the type of each link, the information the target views of the links must show, and so on.

4.4.2 Defining an absolute link

An absolute link allows you to jump from one workspace to another workspace in one click.

The Smart Bank showcase team defined a Logical view to match its individual requirements. This Logical view shows the different channels that are used by the Smart Bank clients when dealing with the bank.

In the Smart Bank showcase menu that displays the status of these channels, all the views are shown. If an alert is visible on one of these branches, a Logical view user must drill down to obtain detailed information in order to understand from which part of the branch the incident comes from.

The process involved

Figure 4-19 shows the source workspace.

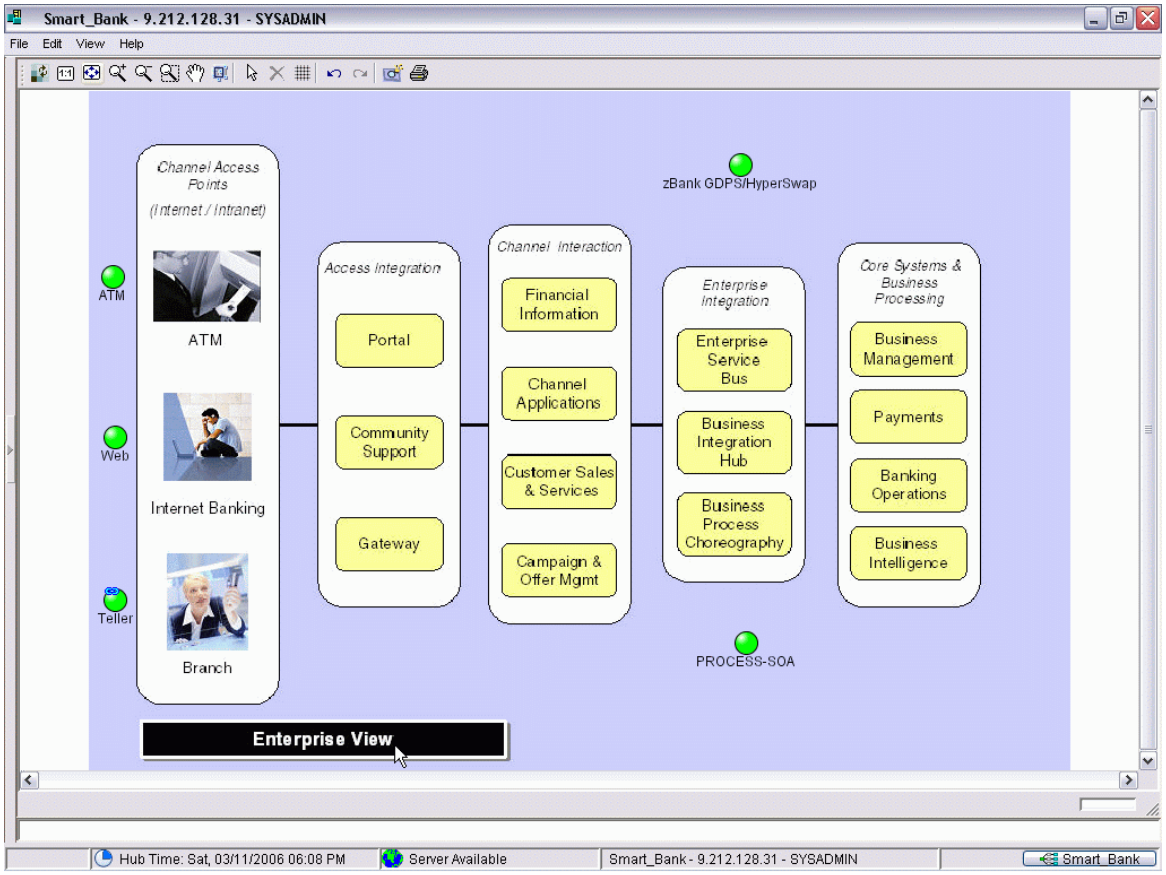


Figure 4-19 Absolute link source workspace: Enterprise view

Figure 4-20 shows the target workspace that is already defined.

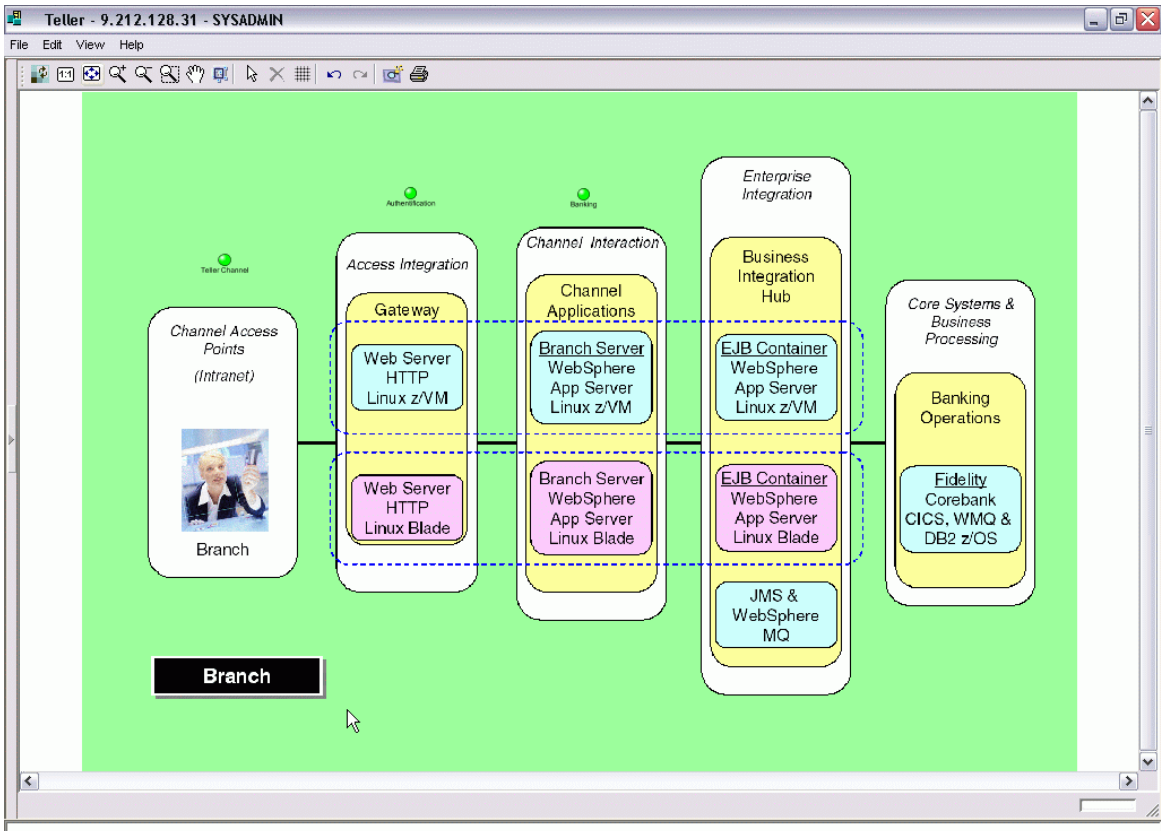


Figure 4-20 Absolute link target workspace: Branch view

The purpose of this new link is to jump from the branch-specific icon to the branch-specific workspace named Teller on the graphic view. To build this link, perform the following tasks:

1. Right-click the icon in the graphic source view. Select **Link to** → **Link Wizard**, as shown in Figure 4-21.

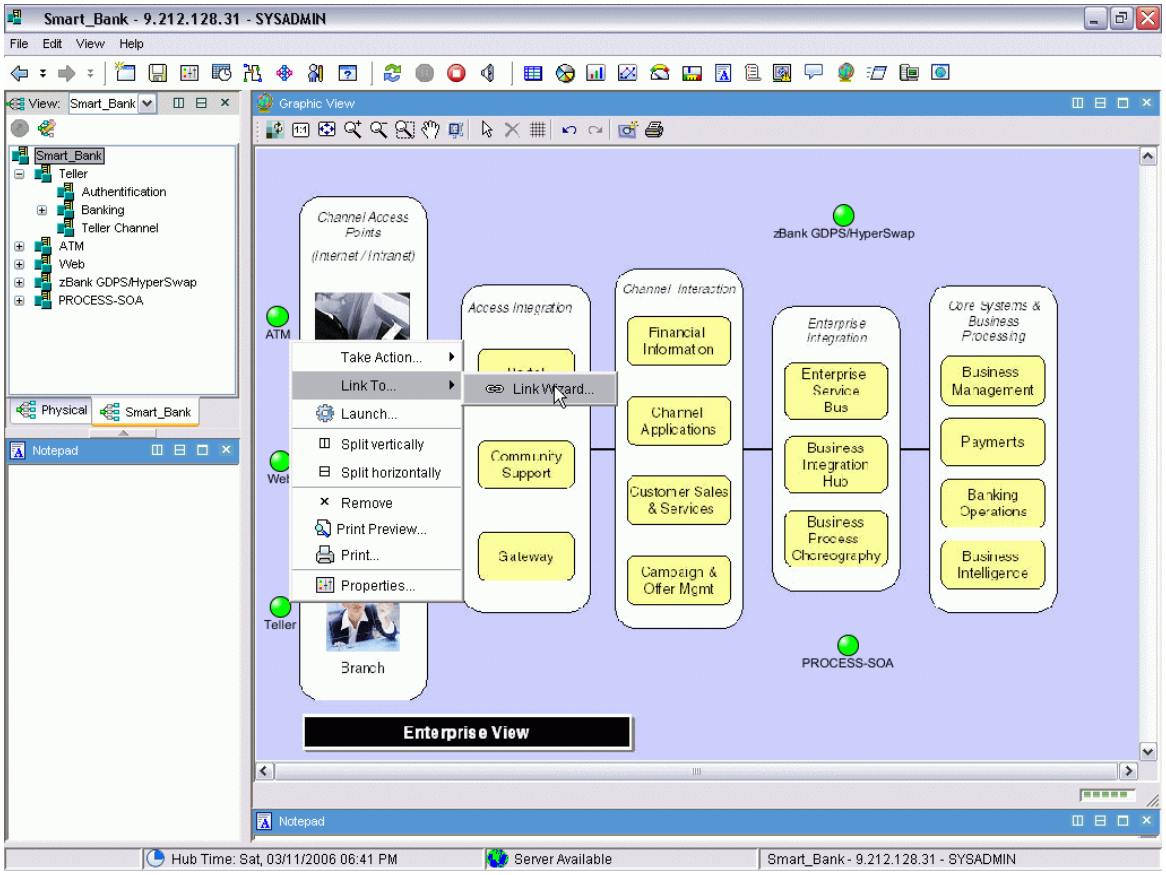


Figure 4-21 Accessing the Link Wizard

2. The TEP now displays the Define New Link window (Figure 4-22). Click **Next**.

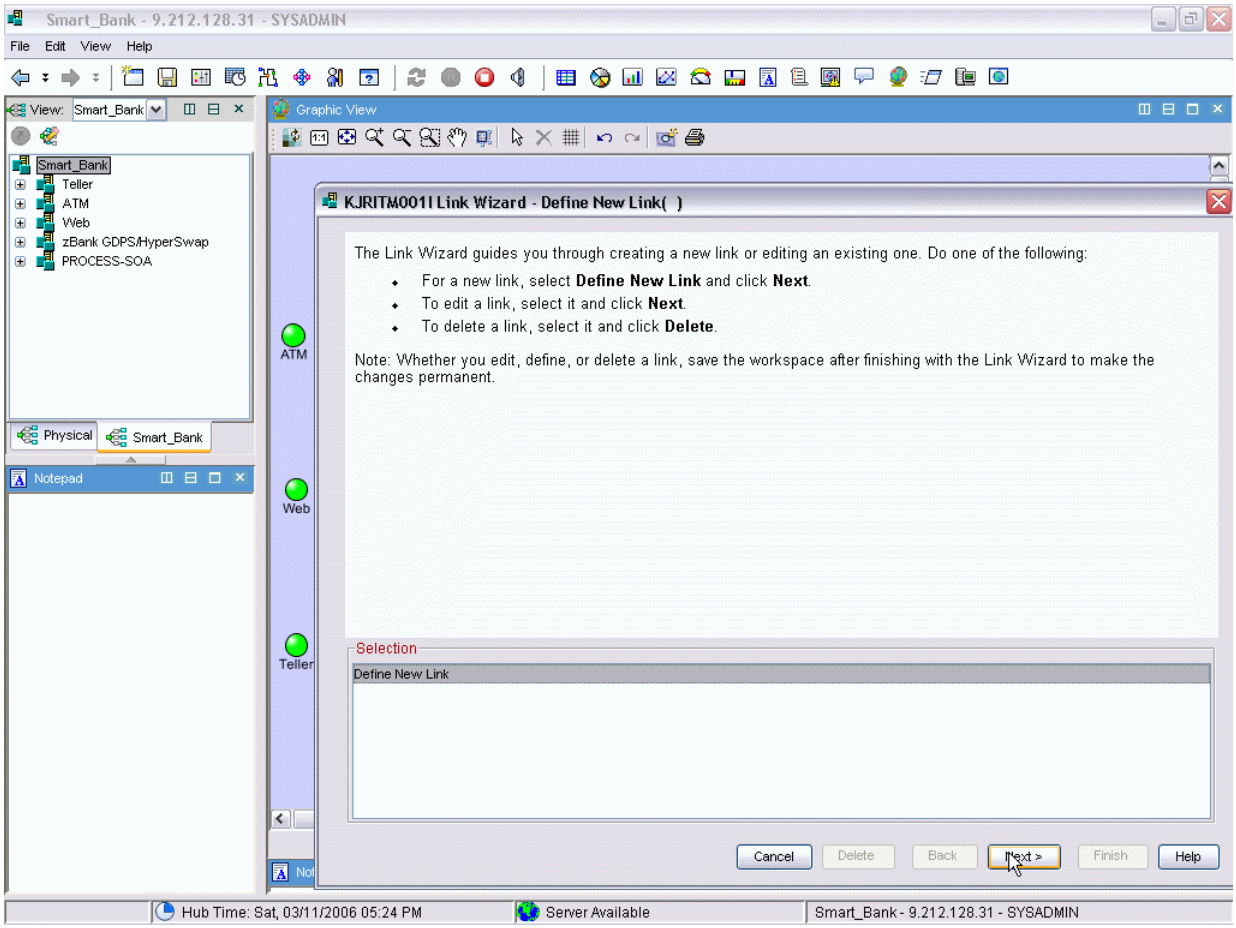


Figure 4-22 Link Wizard access

Note: If links are already defined, the Define New Link window shows the existing definitions.

3. In the next window (Figure 4-23), the Link Identity tag provides a name to your new definition. We recommend that you use standard and meaningful names (blanks are allowed). When the user tries to use the link feature, these names are displayed. If the cursor is positioned on the link symbol, a pop-up displays the name of this link prefixed with the “Link to” label. For this new link, use `Branch details`. When done, click **Next**.

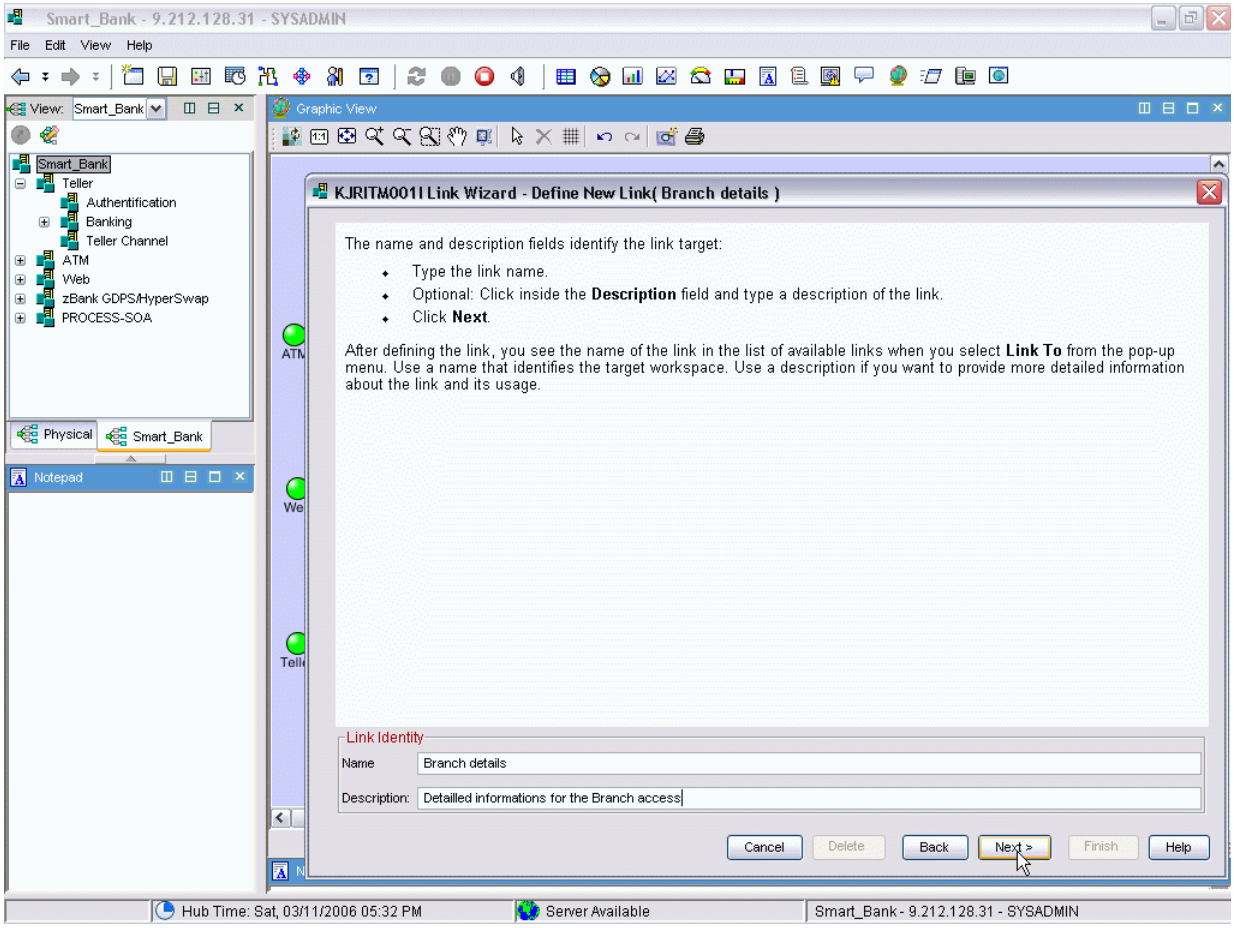

Smart_Bank - 9.212.128.31 - SYSADMIN
File Edit View Help
View: Smart_Bank
Smart_Bank
Teller
Authentification
Banking
Teller Channel
ATM
Web
zBank GDPS/HyperSwap
PROCESS-SOA
Physical
Smart_Bank
Notepad
Graphic View
KJRITM0011 Link Wizard - Define New Link(Branch details)
The name and description fields identify the link target:
Type the link name.
Optional: Click inside the Description field and type a description of the link.
Click Next.
After defining the link, you see the name of the link in the list of available links when you select Link To from the pop-up menu. Use a name that identifies the target workspace. Use a description if you want to provide more detailed information about the link and its usage.
Link Identity
Name
Branch details
Description:
Detailled informations for the Branch access
Cancel
Delete
Back
Next >
Finish
Help
Hub Time: Sat, 03/11/2006 05:32 PM
Server Available
Smart_Bank - 9.212.128.31 - SYSADMIN

Figure 4-23 Link identity: Giving a name to a link

4. The next window (Figure 4-24) displays three areas.
   – In the bottom left (zone A), the Target area allows you to choose the target workspace that will be the target of the link. In the Navigator View, the current Logical view is displayed. Select any other existing view through this prompt.

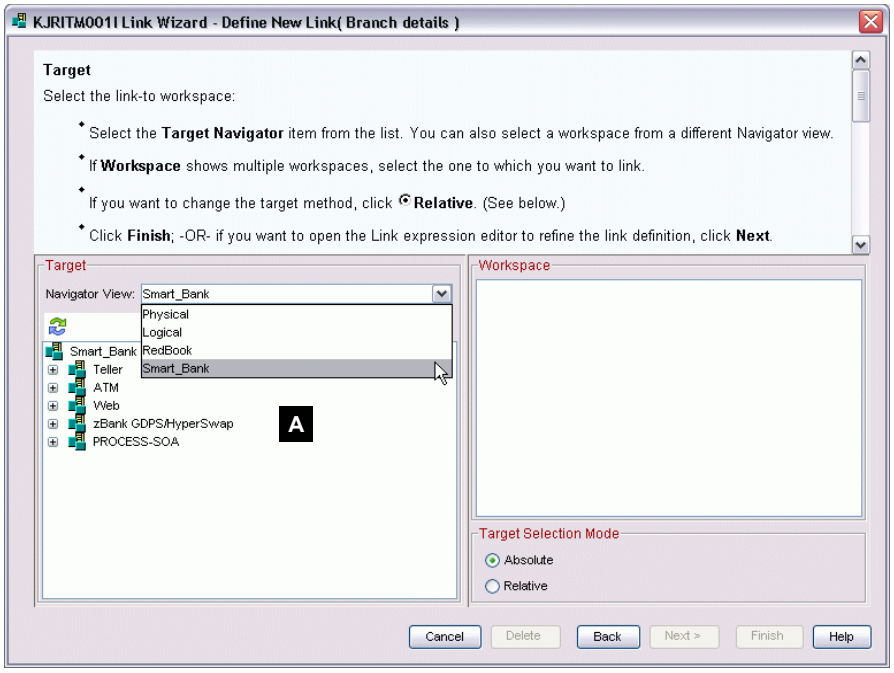


Figure 4-24 Choosing the absolute link target workspace

   – Go to the selected Navigator view (zone A) to reach the target workspace. In our example, we use the Branch workspace (Figure 4-25). Select **Teller**.

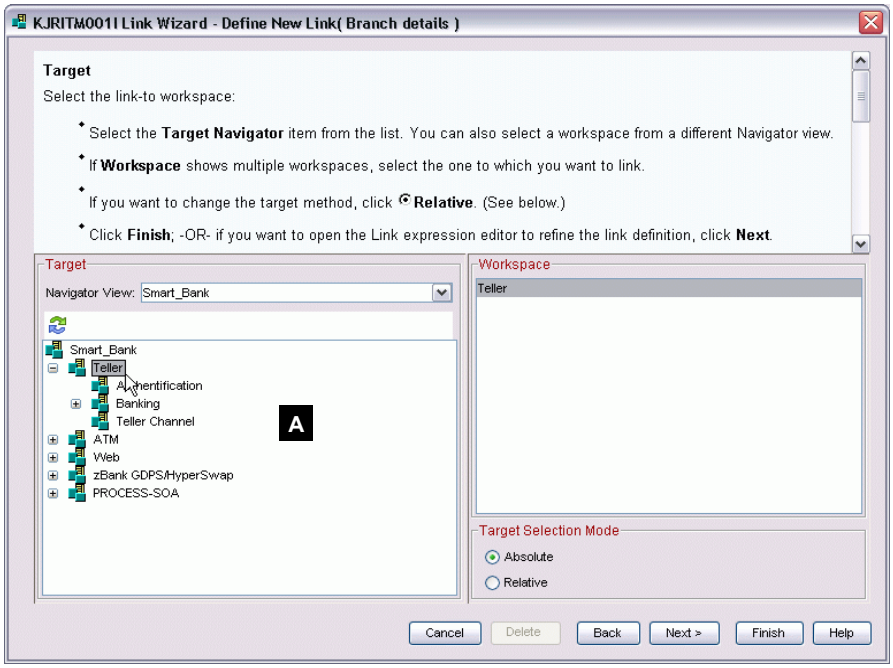


Figure 4-25 Finding the target workspace

– In the area called *Workspace* (zone B, as shown in Figure 4-26), the name of the targeted workspace is displayed. In the area called *Target Selection* mode (zone C) select **Absolute**, and then click **Finish**.

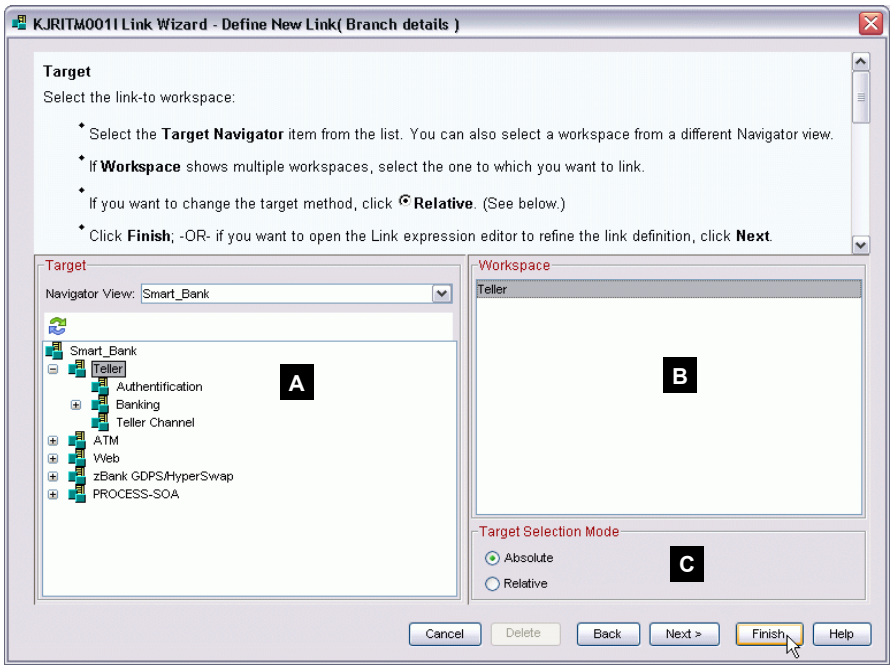


Figure 4-26 Absolute link: End of definition

5. The process of defining an absolute link is complete. To use this new link, right-click the icon on which the definition is set up. The name of the new link is displayed, as shown in Figure 4-27. Click it.

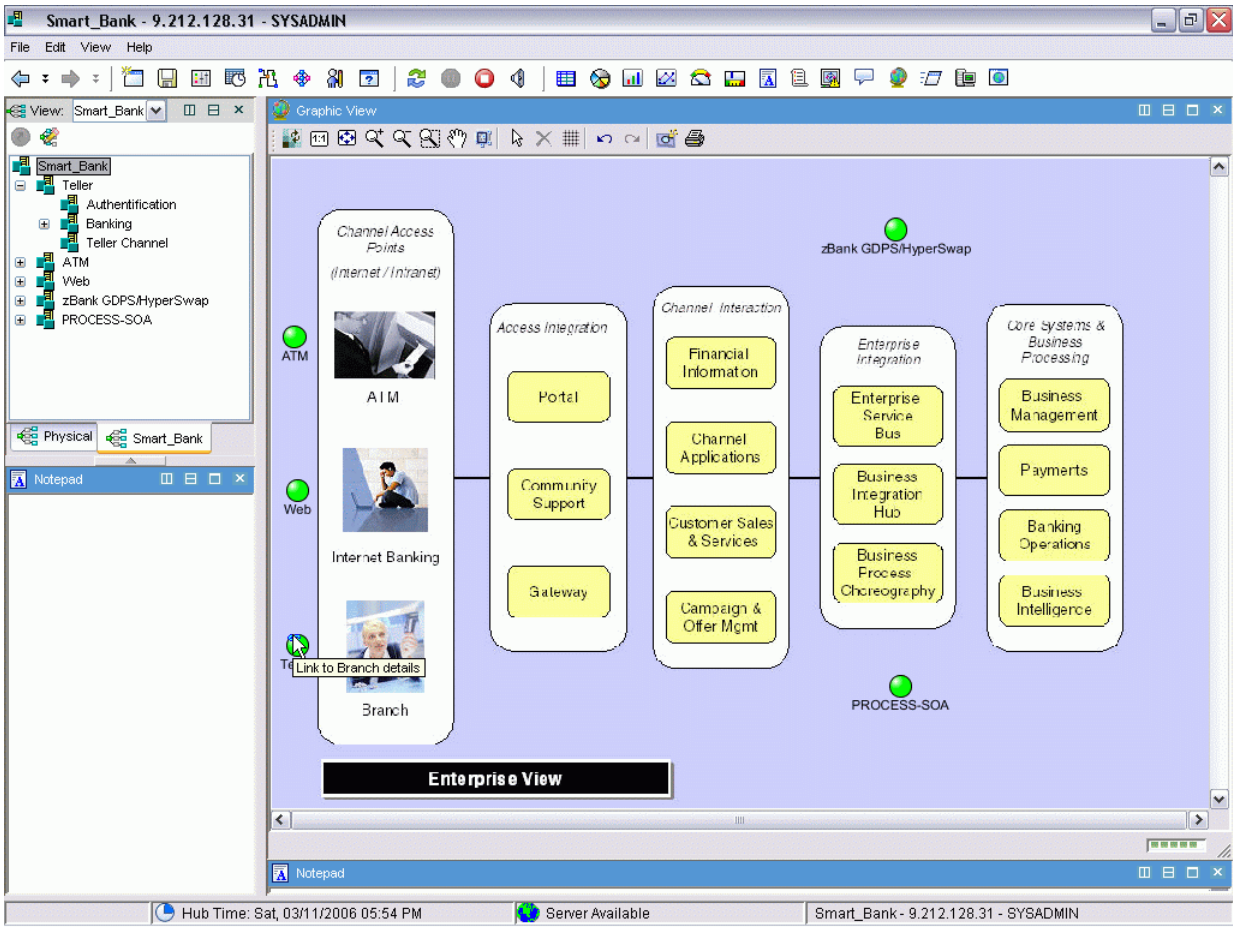


Figure 4-27 Utilization of the new absolute link

6. The TEP executes the demand and the target workspace is now displayed, as shown in Figure 4-28.

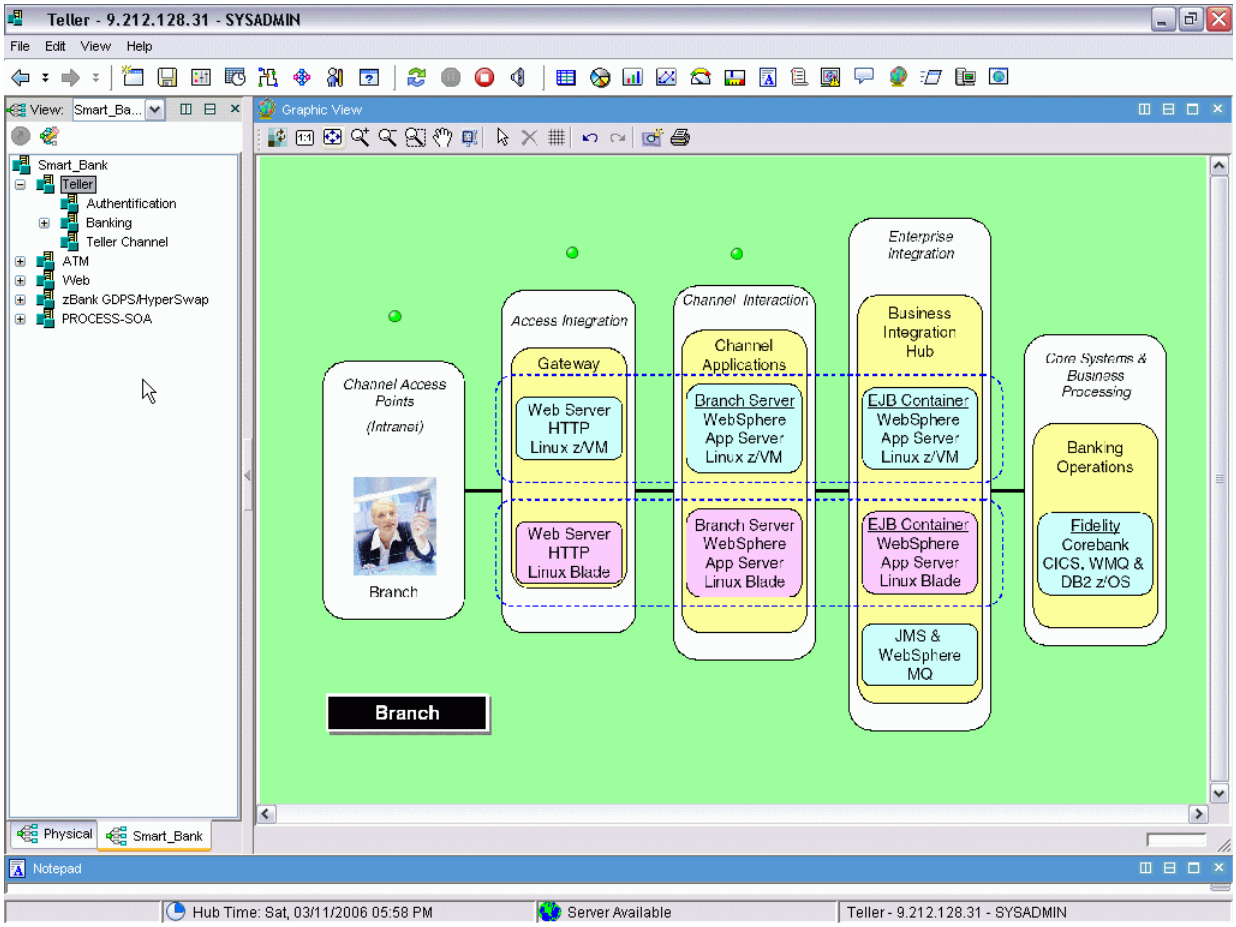


Figure 4-28 Absolute link result

The absolute link that is created helps establish a static connection from one point, for example, the Branch icon inside the status channels view, to another fixed point, for example, the Branch workspace.

4.4.3 Defining a relative link

The purpose of a relative link is to define a link that can be used to target more than one workspace. Some agents monitor more than one resource at a time. As an example, OMEGAMON XE for CICS manages more than one region on a specific z/OS system. When examining the performance of the entire system, a user may want to zoom in on some specific details on a monitored resource, with the possibility of choosing the one that is to be examined in detail. A relative link is designed for that purpose.

The process of defining a relative link is the same as that used to define an absolute link, except that the Relative button is selected when defining the relative link. This link jumps to a chosen subpart of the tree that has the same configuration (managed system type and identical workspace).

Restriction: Generally, a logical tree has no identical subpart. Generally, a TEP administrator must not use a relative link in Logical views.

To define a relative link, perform the following tasks:

1. In the window shown in Figure 4-29, open the CICS part of the Navigation tree on the physical tree. Define a relative link in one of the displayed views. In our case, we define a relative link from a simple table view that displays an overview of each CICS region managed by the Smart Bank showcase team. The target of this relative link is the DB2 Summary workspace available under each controlled CICS region.

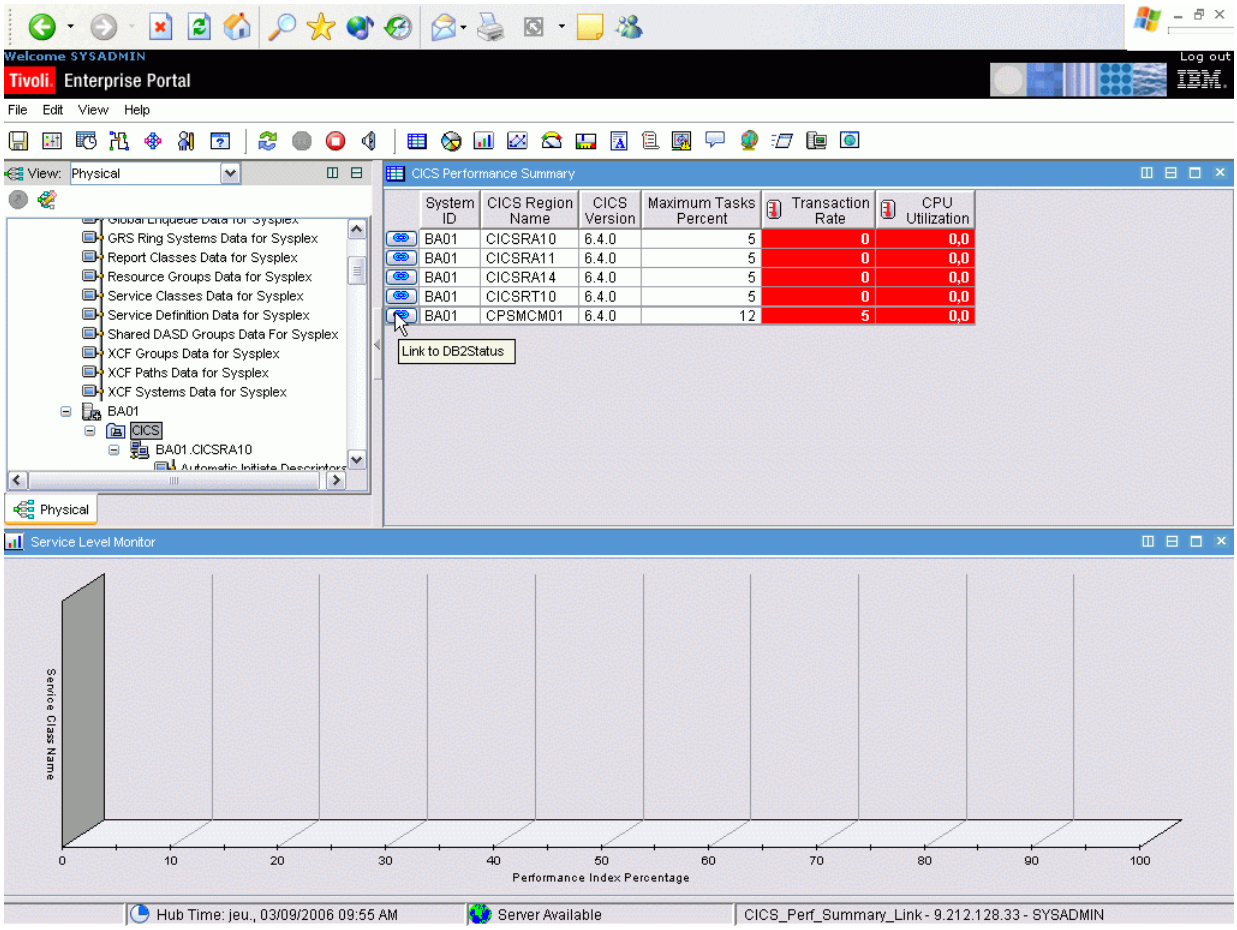


Figure 4-29 Relative link: Sample source view

2. If you use this link by clicking the chain that represents the relative link, the TEP asks you the specific CICS region you want to focus on. Choose the one you want, as shown in Figure 4-30.

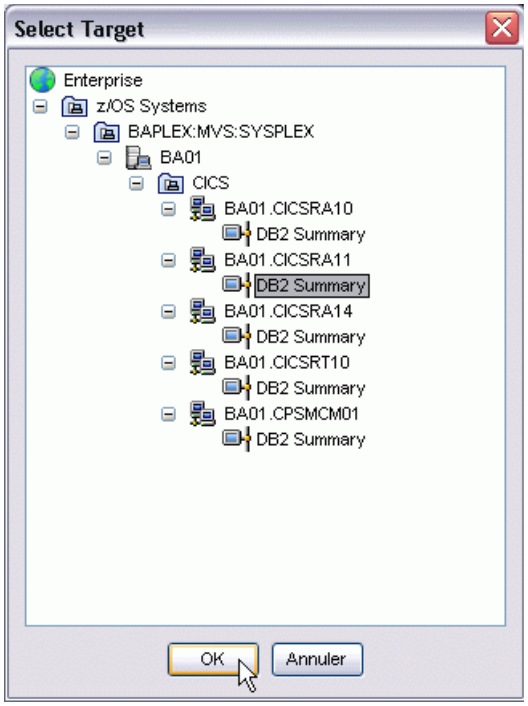


Figure 4-30 The Select Target window when using a relative link

3. The TEP then displays the DB2 Summary workspace for the CICS region that is selected (Figure 4-31).

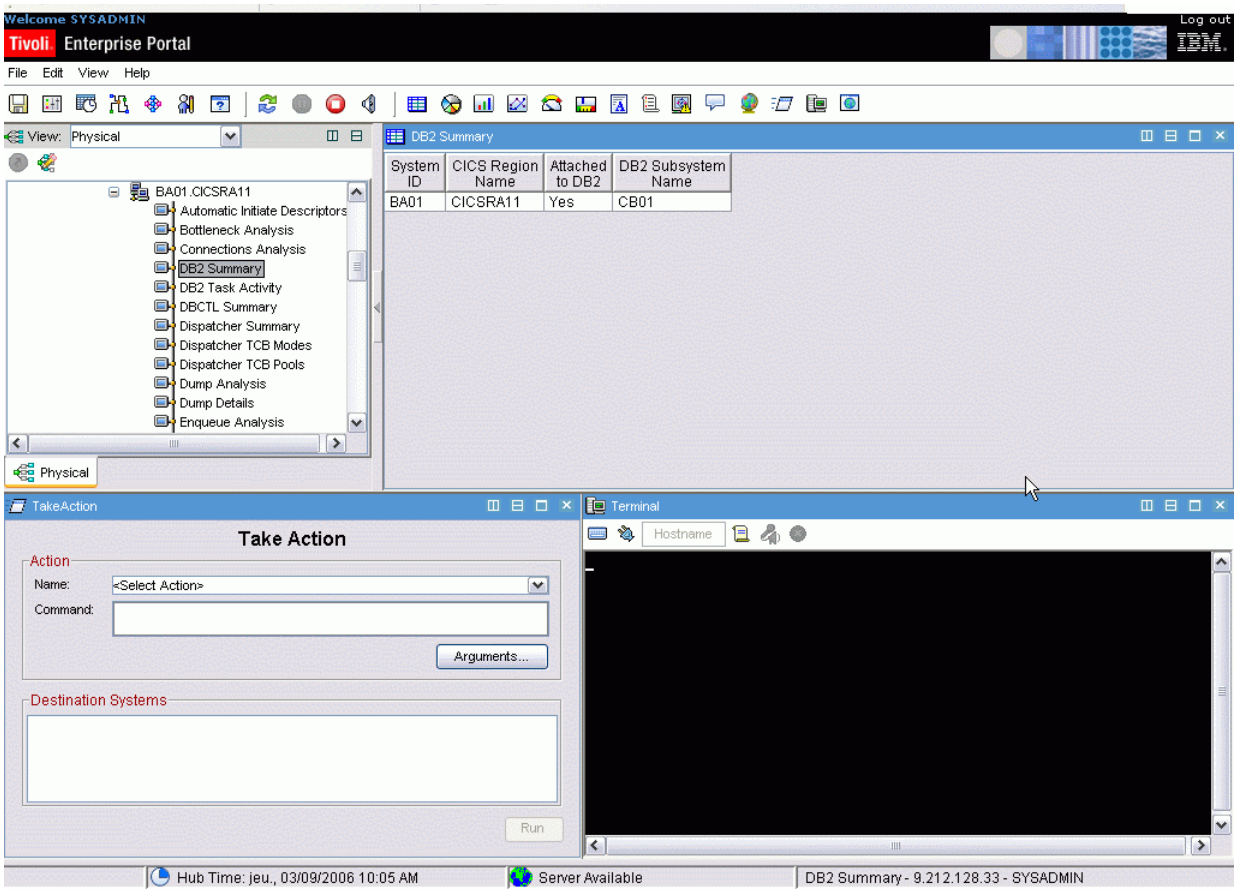


Figure 4-31 Result of a relative link

4.4.4 Defining an advanced link

An advanced link is a powerful tool. It allows you to select data from one view in a workspace and use it to dynamically filter the data on the targeted workspace.

As an example, the Smart Bank showcase team wants to apply specific service levels to its VIP clients. When alerts come in, the team wants to deep-dive into the data pertaining to *only* the VIP clients. A jump from one view in a workspace to another view *with data selection* is the reason behind the dynamic links.

When an advanced link is used, the link takes data from the selected part of the current view (source workspace), goes to the target workspace, and uses the selected data as filters. At the end of the link process, the target workspace shows only the pertinent data relating to the selected data.

To filter the data on the target workspace, the dynamic link function uses variables. These variables, called *symbols*, are set up in the queries used in the target workspace.

Therefore, in order to be operative, the dynamic link function requires the following:

- Internal symbols to determine whether or not to offer a defined link on that place
- Queries in the target workspace set up with symbols
- Association of these symbols with data fields coming from the source workspace
- Expressions using values, functions, and operators to build advanced associations, if necessary

A step-by-step definition

This section describes the definition of an advanced link. It is an example that is built by using OMEGAMON XE for z/OS.

Source workspace for dynamic link definition

The user displays the Workload Manager (WLM) definition on the two z/OS systems used by the Smart Bank showcase team. The workspace called WLM Service Class Resources Workspace is used as a base. In our case, we defined another one for our own use called WLM All Service Class, which is shown in Figure 4-32.

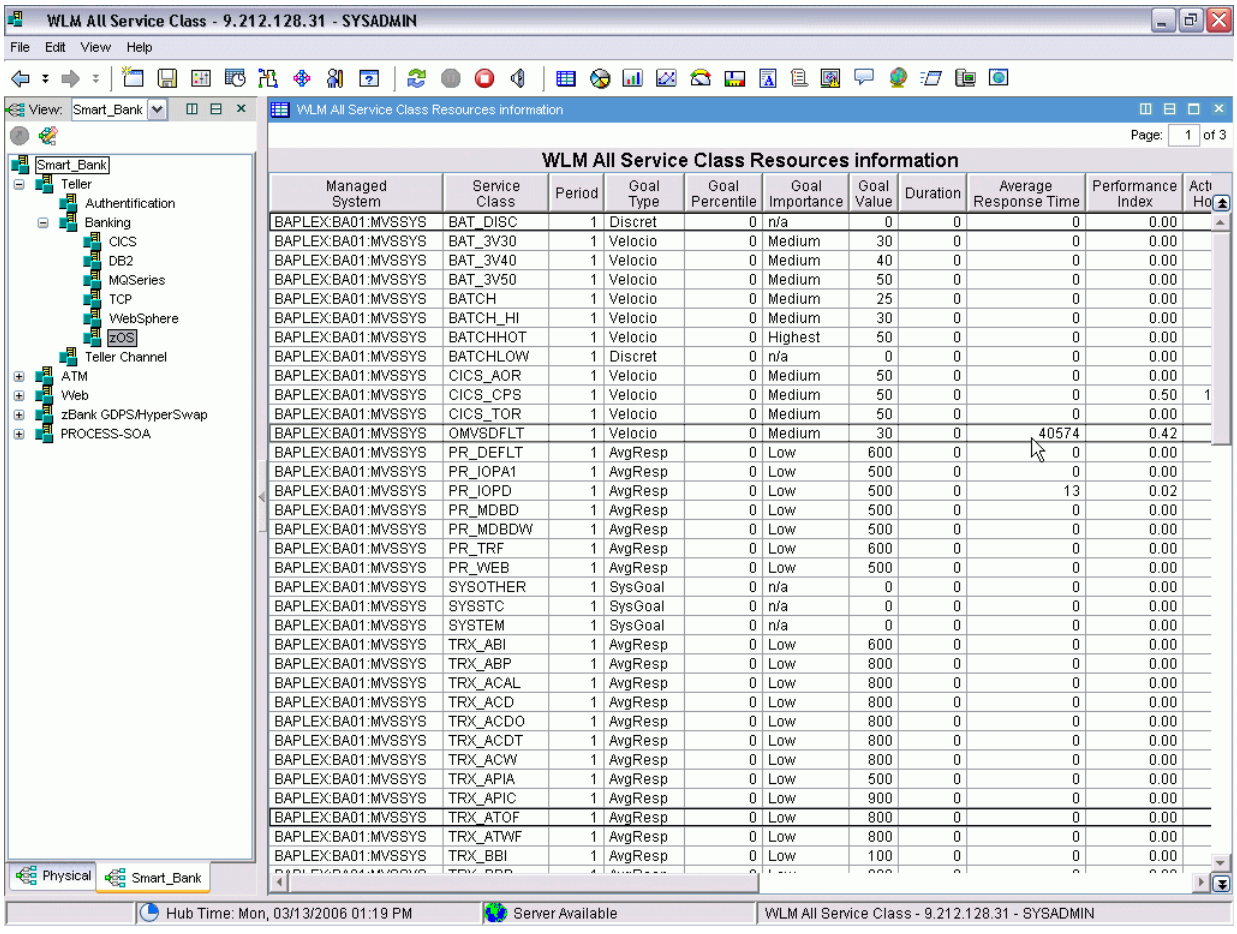

WLM All Service Class - 9.212.128.31 - SYSADMIN
File Edit View Help
View: Smart_Bank
Smart_Bank
Teller
Authentification
Banking
CICS
DB2
MQSeries
TCP
WebSphere
zOS
Teller Channel
ATM
Web
zBank GDPS/HyperSwap
PROCESS-SOA
Physical
Smart_Bank
WLM All Service Class Resources information
Page: 1 of 3
WLM All Service Class Resources information
Managed System | Service Class | Period | Goal Type | Goal Percentile | Goal Importance | Goal Value | Duration | Average Response Time | Performance Index
BAPLEX:BA01:MVSSYS | BAT_DISC | 1 | Discret | 0 | n/a | 0 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | BAT_3V30 | 1 | Velocio | 0 | Medium | 30 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | BAT_3V40 | 1 | Velocio | 0 | Medium | 40 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | BAT_3V50 | 1 | Velocio | 0 | Medium | 50 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | BATCH | 1 | Velocio | 0 | Medium | 25 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | BATCH_HI | 1 | Velocio | 0 | Medium | 30 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | BATCHHOT | 1 | Velocio | 0 | Highest | 50 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | BATCHLOW | 1 | Discret | 0 | n/a | 0 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | CICS_AOR | 1 | Velocio | 0 | Medium | 50 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | CICS_CPS | 1 | Velocio | 0 | Medium | 50 | 0 | 0 | 0.50
BAPLEX:BA01:MVSSYS | CICS_TOR | 1 | Velocio | 0 | Medium | 50 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | OMVSDFLT | 1 | Velocio | 0 | Medium | 30 | 0 | 40574 | 0.42
BAPLEX:BA01:MVSSYS | PR_DEFLT | 1 | AvgResp | 0 | Low | 600 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | PR_IOPA1 | 1 | AvgResp | 0 | Low | 500 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | PR_IOPD | 1 | AvgResp | 0 | Low | 500 | 0 | 13 | 0.02
BAPLEX:BA01:MVSSYS | PR_MDBD | 1 | AvgResp | 0 | Low | 500 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | PR_MDBDW | 1 | AvgResp | 0 | Low | 500 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | PR_TRF | 1 | AvgResp | 0 | Low | 600 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | PR_WEB | 1 | AvgResp | 0 | Low | 500 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | SYSOTHER | 1 | SysGoal | 0 | n/a | 0 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | SYSSTC | 1 | SysGoal | 0 | n/a | 0 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | SYSTEM | 1 | SysGoal | 0 | n/a | 0 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ABI | 1 | AvgResp | 0 | Low | 600 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ABP | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ACAL | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ACD | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ACDO | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ACDT | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ACW | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_APIA | 1 | AvgResp | 0 | Low | 500 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_APIC | 1 | AvgResp | 0 | Low | 900 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ATOF | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_ATWF | 1 | AvgResp | 0 | Low | 800 | 0 | 0 | 0.00
BAPLEX:BA01:MVSSYS | TRX_BBI | 1 | AvgResp | 0 | Low | 100 | 0 | 0 | 0.00
Hub Time: Mon, 03/13/2006 01:19 PM
Server Available
WLM All Service Class - 9.212.128.31 - SYSADMIN

Figure 4-32 Source workspace for dynamic link definition

Objective of the advanced link definition

Because the WLM definitions are numerous, three pages are returned. A user usually selects only one WLM service class and gets back a display showing only the activity for that WLM service class that is active on the two systems at the same time (BA01 and BA02).

Target workspace expected

The content of the target workspace that is expected after the selection is the same as the one called WLM All Service Class. The expected target workspace can be called WLM *Selected* Service Class. In fact, there is another simple way in which to achieve this objective. Simply filter the contents of the view (Figure 4-33). However, in our case, instead of using this repetitive approach (a new filter to focus on each service class), we use the dynamic link function, which performs the filtering function by itself dynamically, after a selection.

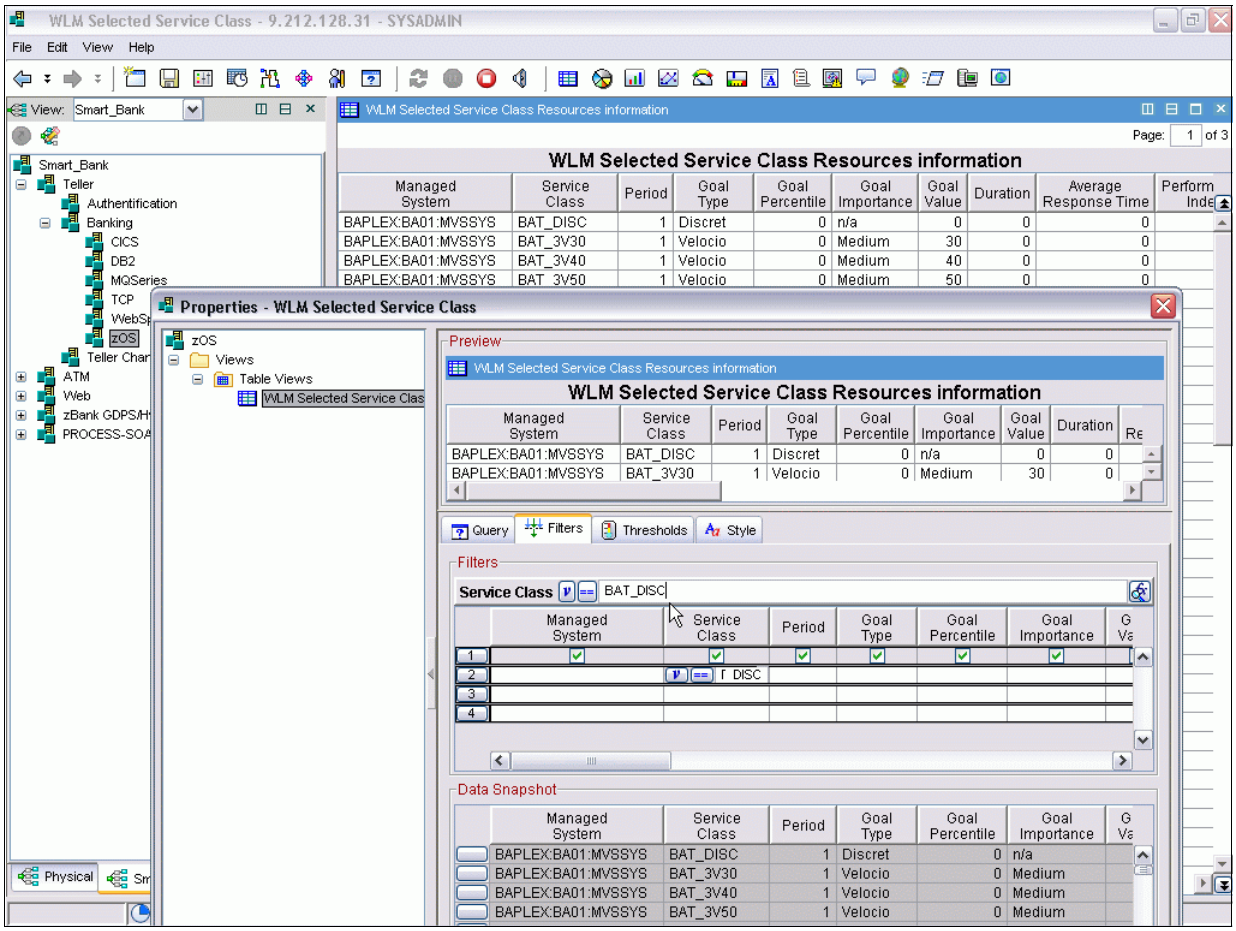
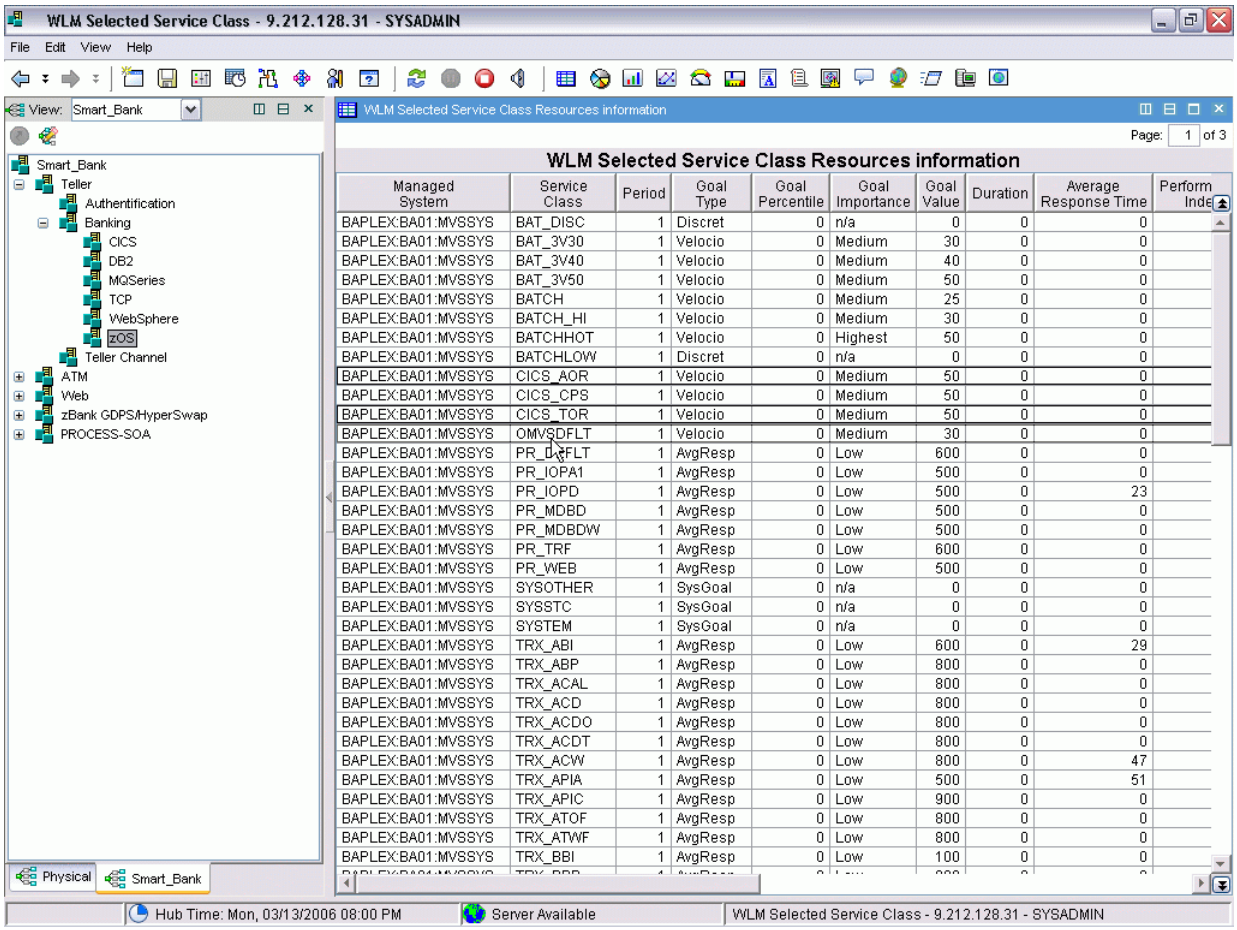


Figure 4-33 Target workspace for dynamic link definition

Defining the target workspace for advanced link definition

To demonstrate this, a simple example is provided. The target workspace, WLM Selected Service Class, includes only the following two elements (Figure 4-34):

- The Logical view navigator
- The view displaying the WLM Service Class



Managed System	Service Class	Period	Goal Type	Goal Percentile	Goal Importance	Goal Value	Duration	Average Response Time	Perform Inde
BAPLEX:BA01:MVSSYS	BAT_DISC	1	Discret	0	n/a	0	0	0	
BAPLEX:BA01:MVSSYS	BAT_3V30	1	Velocio	0	Medium	30	0	0	
BAPLEX:BA01:MVSSYS	BAT_3V40	1	Velocio	0	Medium	40	0	0	
BAPLEX:BA01:MVSSYS	BAT_3V50	1	Velocio	0	Medium	50	0	0	
BAPLEX:BA01:MVSSYS	BATCH	1	Velocio	0	Medium	25	0	0	
BAPLEX:BA01:MVSSYS	BATCH_HI	1	Velocio	0	Medium	30	0	0	
BAPLEX:BA01:MVSSYS	BATCHHOT	1	Velocio	0	Highest	50	0	0	
BAPLEX:BA01:MVSSYS	BATCHLOW	1	Discret	0	n/a	0	0	0	
BAPLEX:BA01:MVSSYS	CICS_AOR	1	Velocio	0	Medium	50	0	0	
BAPLEX:BA01:MVSSYS	CICS_CPS	1	Velocio	0	Medium	50	0	0	
BAPLEX:BA01:MVSSYS	CICS_TOR	1	Velocio	0	Medium	50	0	0	
BAPLEX:BA01:MVSSYS	OMVSDFLT	1	Velocio	0	Medium	30	0	0	
BAPLEX:BA01:MVSSYS	PR_DEFLT	1	AvgResp	0	Low	600	0	0	
BAPLEX:BA01:MVSSYS	PR_IOPA1	1	AvgResp	0	Low	500	0	0	
BAPLEX:BA01:MVSSYS	PR_IOPD	1	AvgResp	0	Low	500	0	23	
BAPLEX:BA01:MVSSYS	PR_MDBD	1	AvgResp	0	Low	500	0	0	
BAPLEX:BA01:MVSSYS	PR_MDBDW	1	AvgResp	0	Low	500	0	0	
BAPLEX:BA01:MVSSYS	PR_TRF	1	AvgResp	0	Low	600	0	0	
BAPLEX:BA01:MVSSYS	PR_WEB	1	AvgResp	0	Low	500	0	0	
BAPLEX:BA01:MVSSYS	SYSOTHER	1	SysGoal	0	n/a	0	0	0	
BAPLEX:BA01:MVSSYS	SYSSTC	1	SysGoal	0	n/a	0	0	0	
BAPLEX:BA01:MVSSYS	SYSTEM	1	SysGoal	0	n/a	0	0	0	
BAPLEX:BA01:MVSSYS	TRX_ABI	1	AvgResp	0	Low	600	0	29	
BAPLEX:BA01:MVSSYS	TRX_ABP	1	AvgResp	0	Low	800	0	0	
BAPLEX:BA01:MVSSYS	TRX_ACAL	1	AvgResp	0	Low	800	0	0	
BAPLEX:BA01:MVSSYS	TRX_ACD	1	AvgResp	0	Low	800	0	0	
BAPLEX:BA01:MVSSYS	TRX_ACDO	1	AvgResp	0	Low	800	0	0	
BAPLEX:BA01:MVSSYS	TRX_ACDT	1	AvgResp	0	Low	800	0	0	
BAPLEX:BA01:MVSSYS	TRX_ACW	1	AvgResp	0	Low	800	0	47	
BAPLEX:BA01:MVSSYS	TRX_APIA	1	AvgResp	0	Low	500	0	51	
BAPLEX:BA01:MVSSYS	TRX_APIC	1	AvgResp	0	Low	900	0	0	
BAPLEX:BA01:MVSSYS	TRX_ATOF	1	AvgResp	0	Low	800	0	0	
BAPLEX:BA01:MVSSYS	TRX_ATWF	1	AvgResp	0	Low	800	0	0	
BAPLEX:BA01:MVSSYS	TRX_BBI	1	AvgResp	0	Low	100	0	0	

Figure 4-34 The target workspace (without any filtering)

A user query is defined to gather the data. This query is another copy of an existing one called WLM Service Class Resource. Call this query `WLM LINK Service Class Resources` and modify it by adding a *symbol* named `$SERCLASS$` in the Service Class column (zone A in Figure 4-35). Implement this new symbol in this column because the user request is to make the selection based on the Service Class name.

Important: *Never* modify an existing symbol in a query. Changing the name of an existing symbol results in unpredictable failures and errors. (An existing symbol may be used for purposes pertaining to the product, by other links, and so on.)

Note: You can define your own queries in the TEP. That way, you can set up your own symbols and retrieve only the data you require for the views in the workspace.

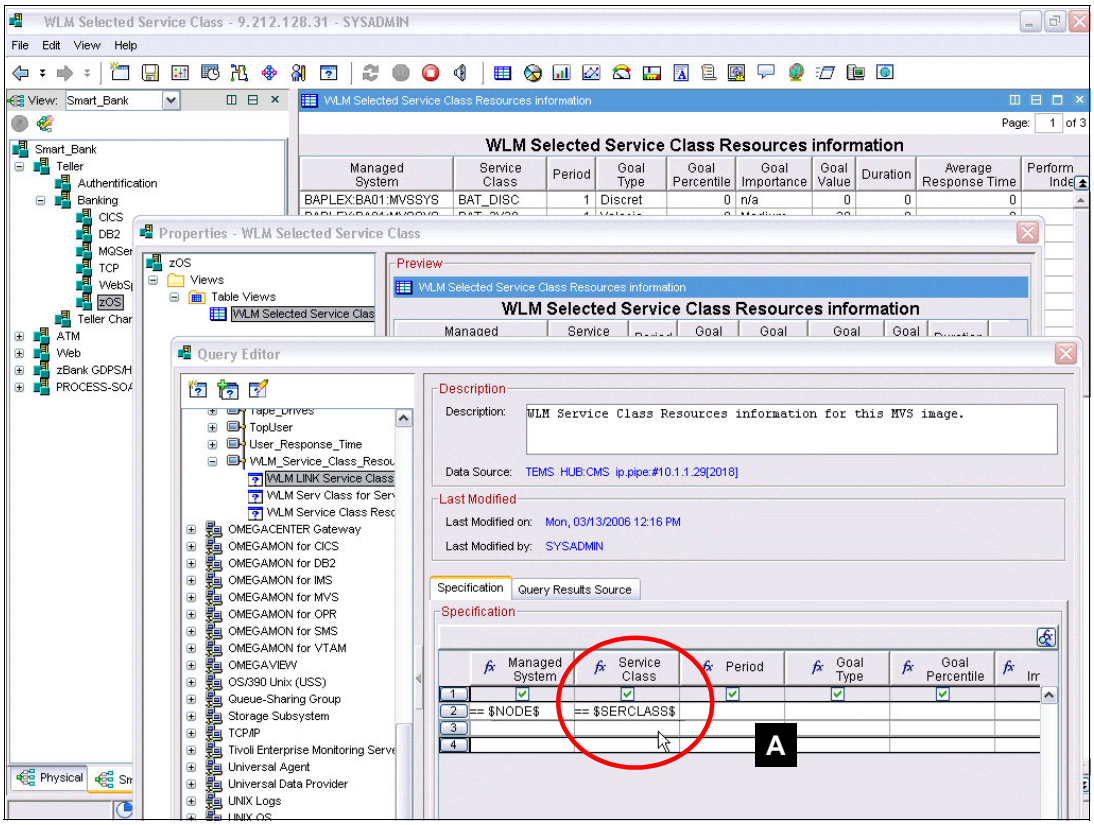


Figure 4-35 Adding a symbol in a query

A new workspace is defined. It contains the Navigator tree and one view. A new query is defined, as also a new symbol in it.

Accessing the advanced link definition

To access the advanced link definition, perform the following tasks:

1. In the workspace named WLM All Service Class, select a line from the table and right-click it.
2. Select the **Link To** → **Link Wizard** option (Figure 4-36).

> **Important:** Make a note of the Service Class name displayed in the line that you select in order to start the link definition. This name is used later in this chapter.

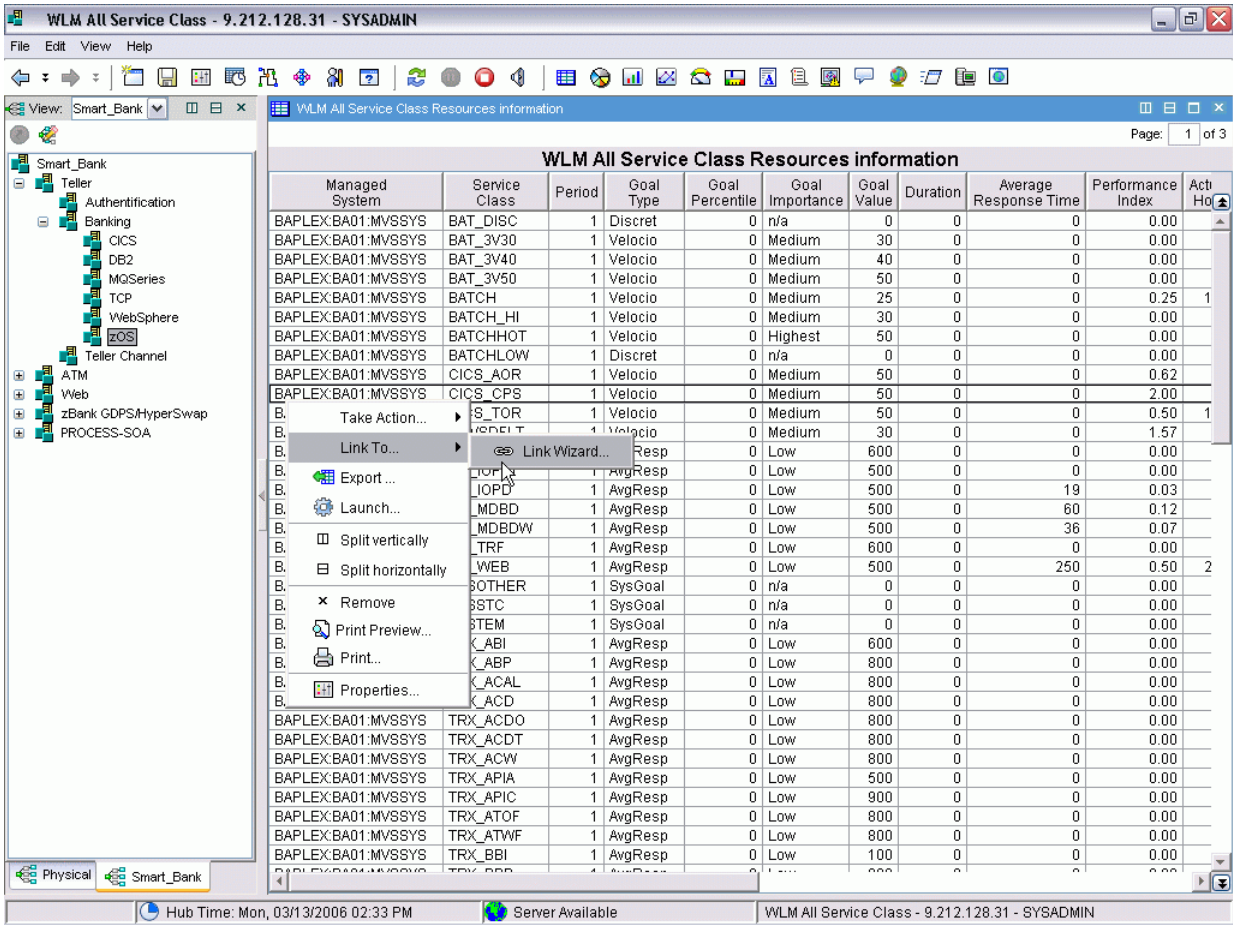

WLM All Service Class - 9.212.128.31 - SYSADMIN
File Edit View Help
View: Smart_Bank
Smart_Bank
Teller
Authentification
Banking
CICS
DB2
MQSeries
TCP
WebSphere
zOS
Teller Channel
ATM
Web
zBank GDPS/HyperSwap
PROCESS-SOA
Physical
Smart_Bank
WLM All Service Class Resources information
Page: 1 of 3
WLM All Service Class Resources information
Managed System
Service Class
Period
Goal Type
Goal Percentile
Goal Importance
Goal Value
Duration
Average Response Time
Performance Index
BAPLEX:BA01:MVSSYS BAT_DISC 1 Discret 0 n/a 0 0 0 0.00
BAPLEX:BA01:MVSSYS BAT_3V30 1 Velocio 0 Medium 30 0 0 0.00
BAPLEX:BA01:MVSSYS BAT_3V40 1 Velocio 0 Medium 40 0 0 0.00
BAPLEX:BA01:MVSSYS BAT_3V50 1 Velocio 0 Medium 50 0 0 0.00
BAPLEX:BA01:MVSSYS BATCH 1 Velocio 0 Medium 25 0 0 0.25
BAPLEX:BA01:MVSSYS BATCH_HI 1 Velocio 0 Medium 30 0 0 0.00
BAPLEX:BA01:MVSSYS BATCHHOT 1 Velocio 0 Highest 50 0 0 0.00
BAPLEX:BA01:MVSSYS BATCHLOW 1 Discret 0 n/a 0 0 0 0.00
BAPLEX:BA01:MVSSYS CICS_AOR 1 Velocio 0 Medium 50 0 0 0.62
BAPLEX:BA01:MVSSYS CICS_CPS 1 Velocio 0 Medium 50 0 0 2.00
Take Action...
Link To...
Export ...
Launch...
Split vertically
Split horizontally
Remove
Print Preview...
Print...
Properties...
Link Wizard...
BAPLEX:BA01:MVSSYS TRX_ACDO 1 AvgResp 0 Low 800 0 0 0.00
BAPLEX:BA01:MVSSYS TRX_ACDT 1 AvgResp 0 Low 800 0 0 0.00
BAPLEX:BA01:MVSSYS TRX_ACW 1 AvgResp 0 Low 800 0 0 0.00
BAPLEX:BA01:MVSSYS TRX_APIA 1 AvgResp 0 Low 500 0 0 0.00
BAPLEX:BA01:MVSSYS TRX_APIC 1 AvgResp 0 Low 900 0 0 0.00
BAPLEX:BA01:MVSSYS TRX_ATOF 1 AvgResp 0 Low 800 0 0 0.00
BAPLEX:BA01:MVSSYS TRX_ATWF 1 AvgResp 0 Low 800 0 0 0.00
BAPLEX:BA01:MVSSYS TRX_BBI 1 AvgResp 0 Low 100 0 0 0.00
Hub Time: Mon, 03/13/2006 02:33 PM
Server Available
WLM All Service Class - 9.212.128.31 - SYSADMIN

Figure 4-36 Accessing the Link Wizard

3. A new link named WLM Details is created, as shown in Figure 4-37. Clicking the **Next** button takes you to the next window asking for the target workspace of the link.

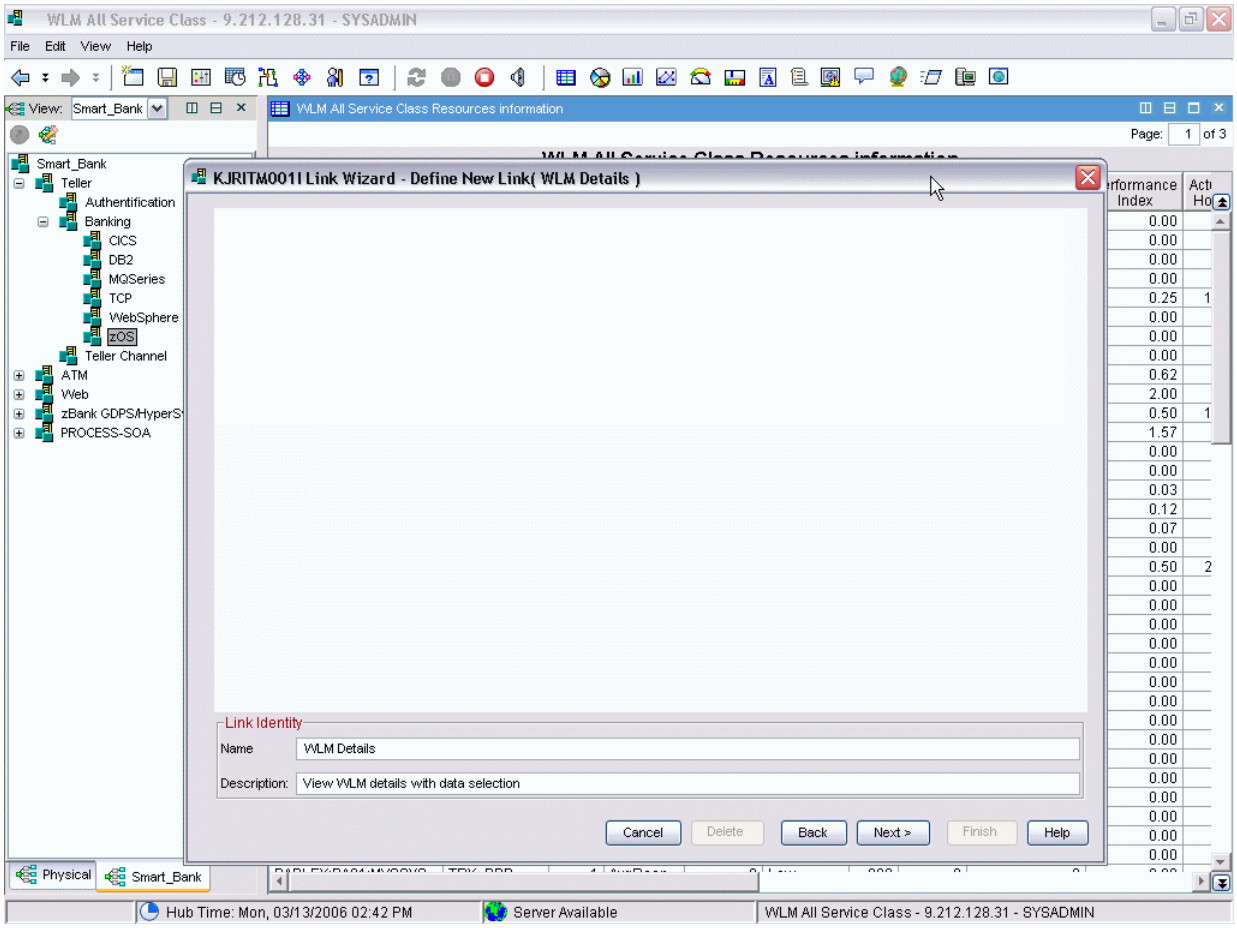


Figure 4-37 Giving a name to the new dynamic link

4. From the Logical view navigator at the z/OS level in our Navigator, select the newly defined workspace called **WLM Selected Service Class**, as shown in Figure 4-38. Instead of clicking **Finish**, as is done for the absolute link or the relative link, click **Next**.

> **Note:** In this case, a link that jumps from one workspace to another workspace defined at the *same* level in the logical tree is defined. There is no rule that states that such a definition must not be performed.

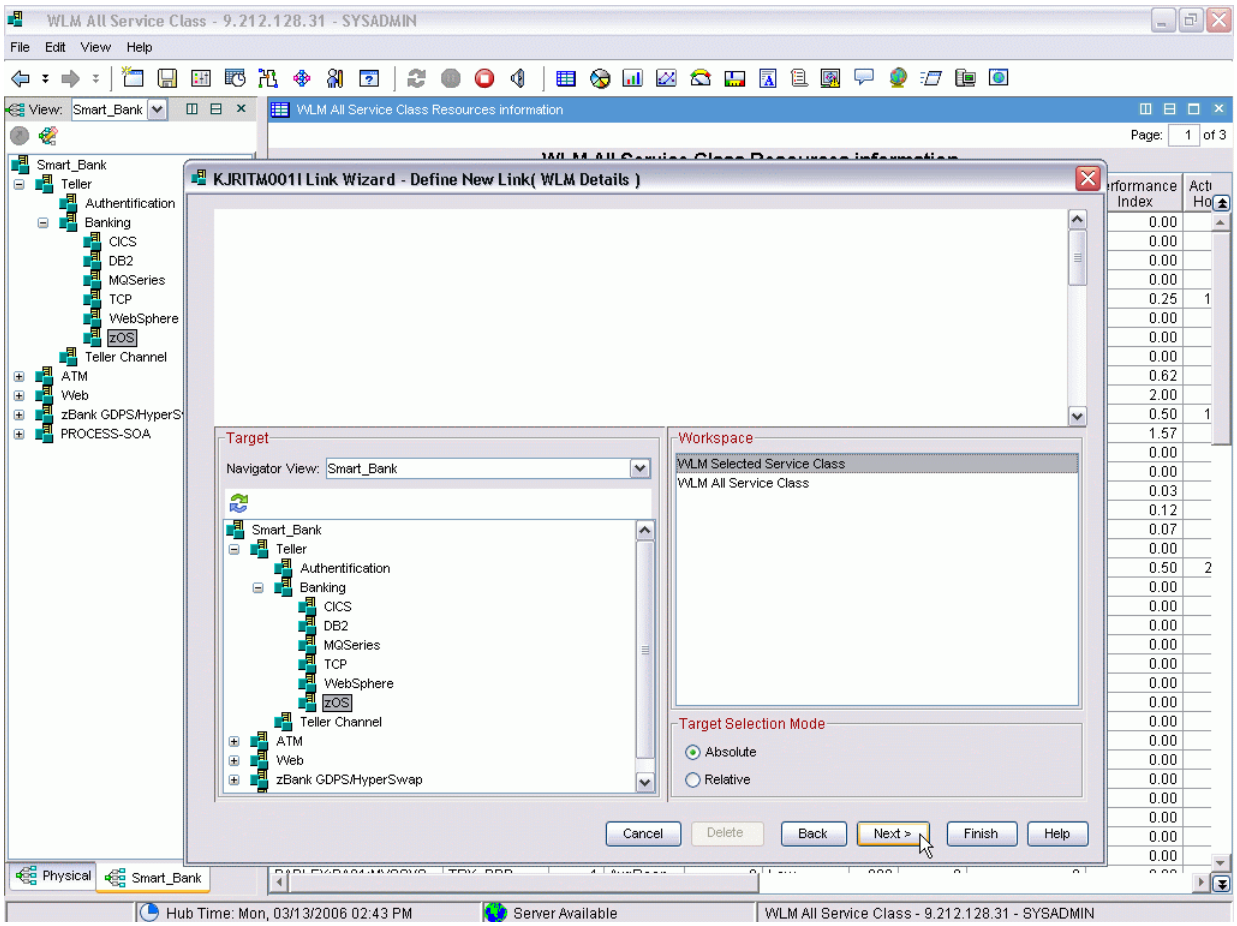


Figure 4-38 Target workspace selection

5. A new panel called Link Expression® Editor is displayed (Figure 4-39). The Helps are also displayed. To close the Help window, click any field in the left side of the window marked with a bold blue “v” before it.

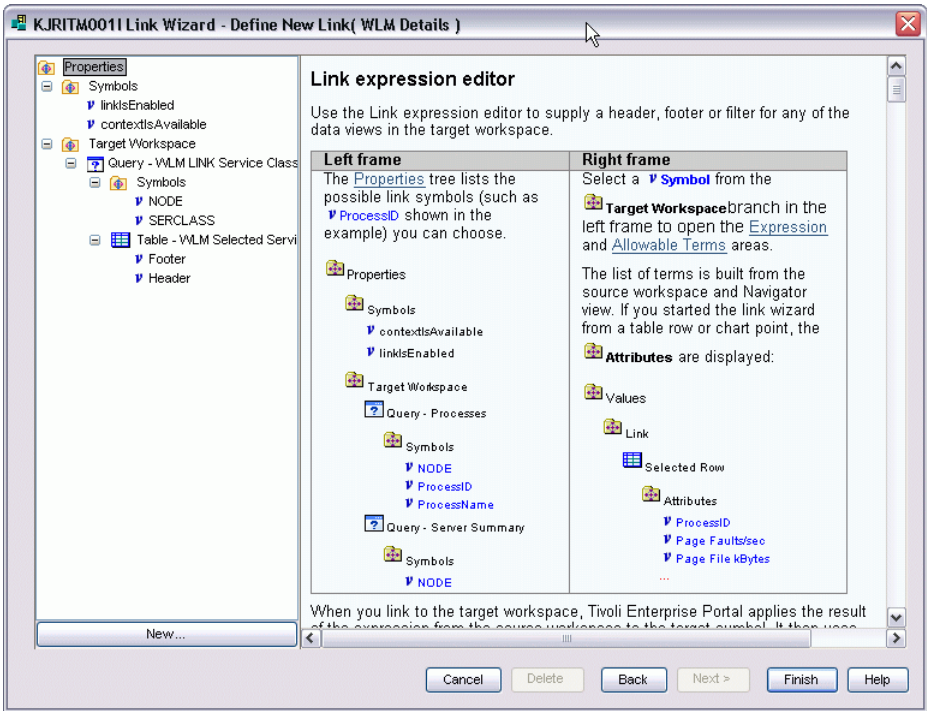

KJRITM0011 Link Wizard - Define New Link(WLM Details)
Properties
Symbols
linkIsEnabled
contextIsAvailable
Target Workspace
Query - WLM LINK Service Class
Symbols
NODE
SERCLASS
Table - WLM Selected Servi
Footer
Header
New...
Link expression editor
Use the Link expression editor to supply a header, footer or filter for any of the data views in the target workspace.
Left frame
Right frame
The Properties tree lists the possible link symbols (such as ProcessID shown in the example) you can choose.
Select a Symbol from the Target Workspace branch in the left frame to open the Expression and Allowable Terms areas.
The list of terms is built from the source workspace and Navigator view. If you started the link wizard from a table row or chart point, the Attributes are displayed:
When you link to the target workspace, Tivoli Enterprise Portal applies the result
Cancel
Delete
Back
Next >
Finish
Help

Figure 4-39 Link Expression Editor panel

The window then displays data in the three parts (Figure 4-40).

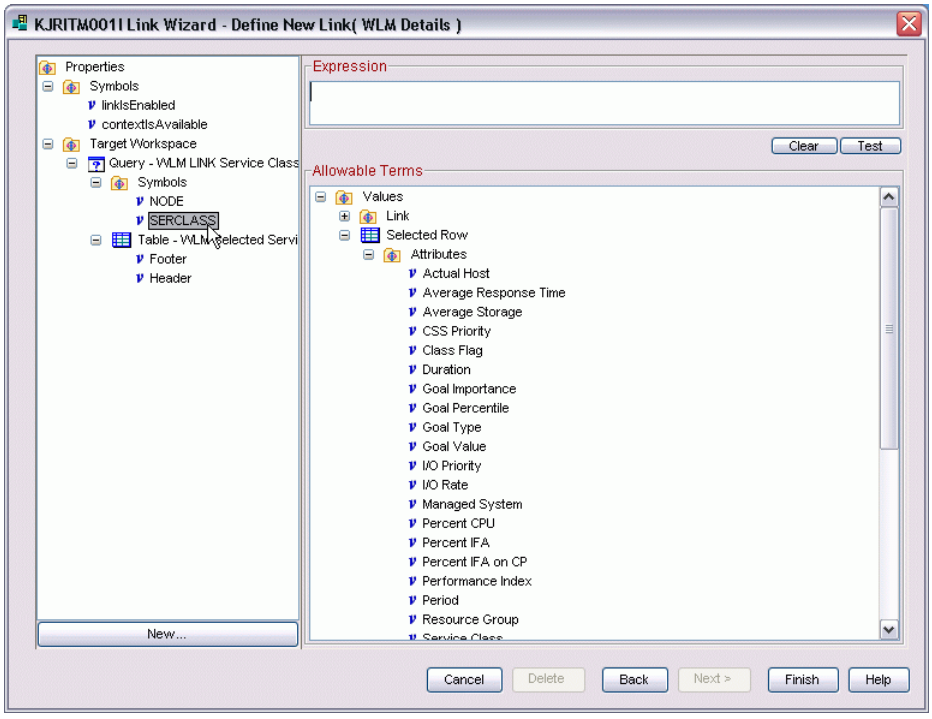


Figure 4-40 The dynamic link definition window

The Advanced Link internal setup

The Define New Link window is divided into the following parts, as shown in Figure 4-41:

- The target workspace content (A) describes the internal content of the target workspace.
- The allowed terms (B) displays values coming from the source workspace, and the functions and operators you can apply on these values.
- The Expression part (C) shows the contents of the values or the results of an expression.

Each of these windows is used during the dynamic link definition.

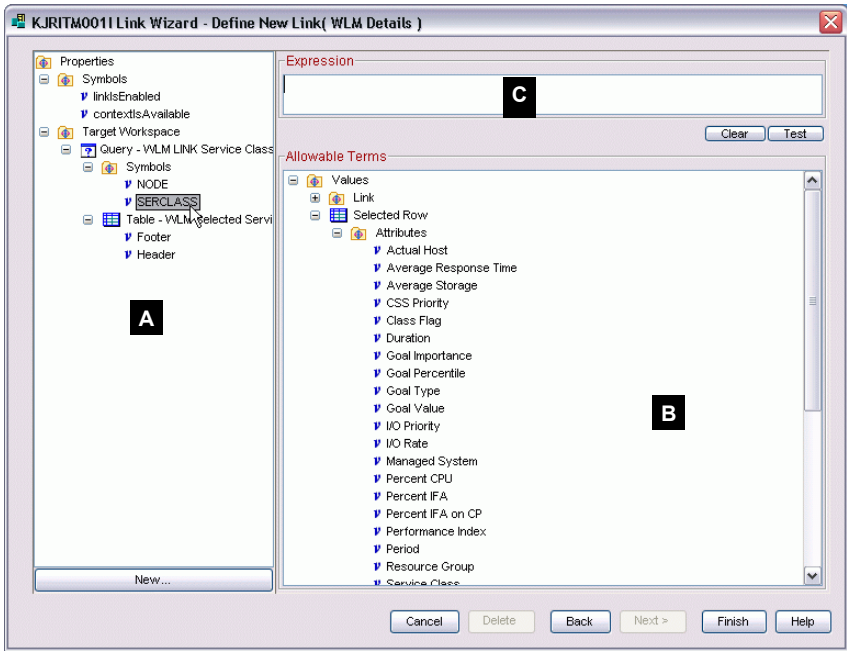


Figure 4-41 Parts of the dynamic link definition window

Advanced link: The target workspace content

The target workspace displays the Navigator tree (zone A) and a table view (B), as shown in Figure 4-42. As stated earlier, in order to create this table view, a query named WLM LINK Service Class Resources (zone C) is used. We have two symbols in this query, $NODE$ and $SERCLASS$ (D).

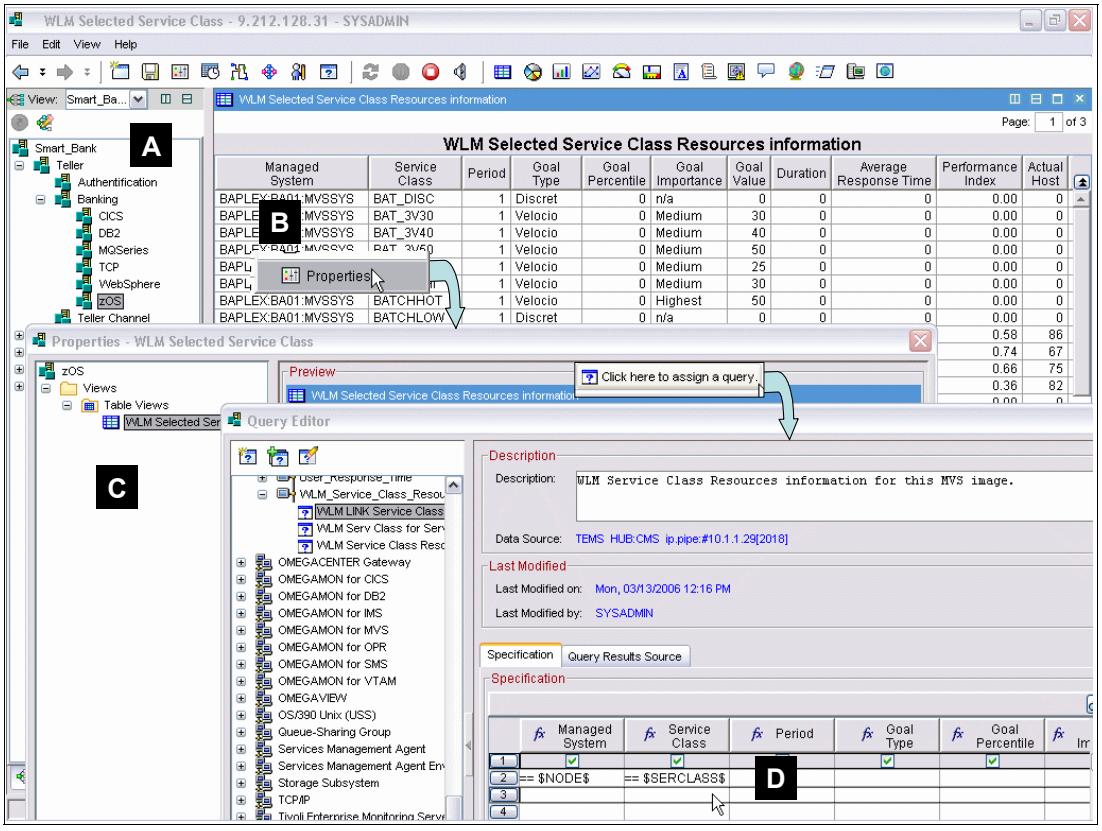


Figure 4-42 The target workspace with a view and its query

Advanced link: Matching the elements together

The target simple workspace is already known (Figure 4-34 on page 217). During the dynamic link definition process, a window shows a display with three parts (Figure 4-41).

In fact, there is a direct association between what the dynamic link definition window displays and the content of the workspace. By bringing the two windows closer, as shown in Figure 4-43, the following points become clear:

- The symbols used in the query in the target workspace (extract in A1) are displayed in the dynamic link window (A2).
- The table, which is the only view in the target workspace (B1), is shown in the dynamic link window (B2), but as a header and footer only (no data shown).
- The dynamic link context and enable (C1) is used by the TEP internally. It uses these values to either offer or not offer the link on that row in that view (true or false values shown in C2).
- Data coming from the source workspace is also displayed (D1).

The structure of the target workspace is completely displayed in the left side of the dynamic link definition window. If multiple views are available in this workspace, this window displays all of them.

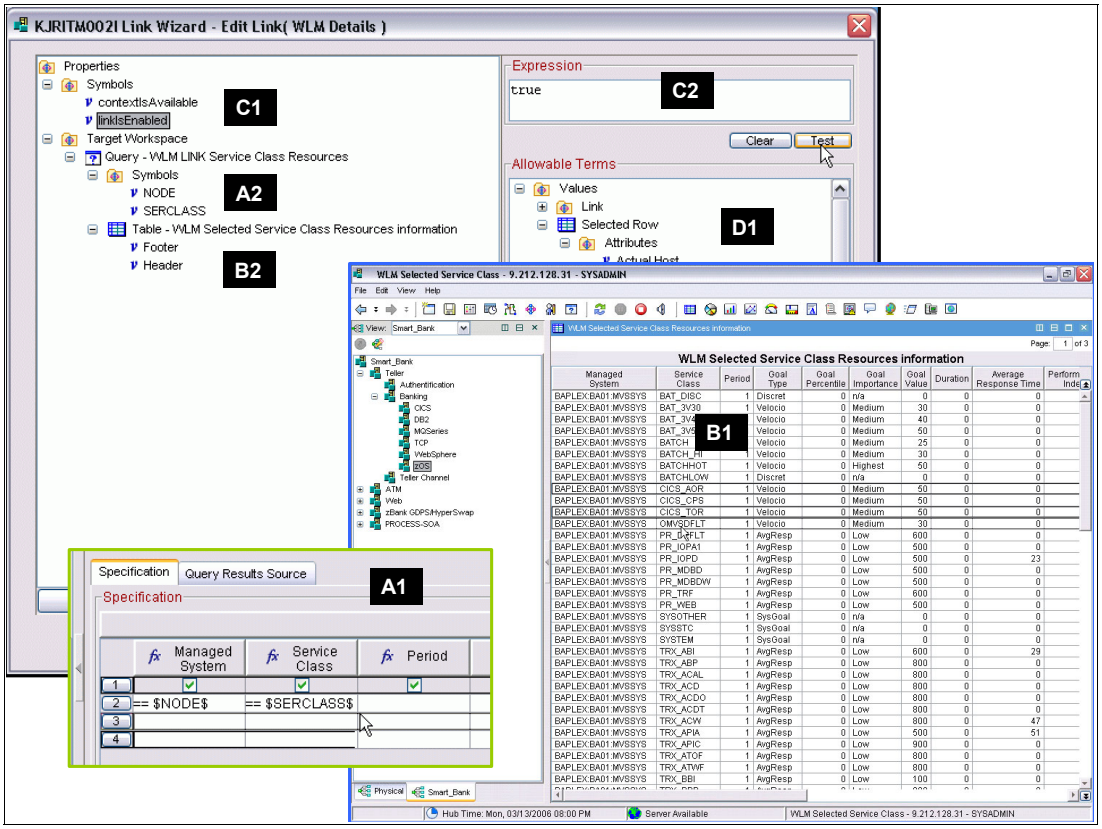


Figure 4-43 The define link window close to the workspace window

The purpose of an advanced link is to filter and to modify the content of the target workspace. To set up the filter value, access to the data displayed by the selected view on the source workspace is required. This data is displayed in the bottom of the dynamic link definition window (Figure 4-44) on the right-hand side. All the attributes gathered by the query attached to the view (and either displayed or not displayed in this view, as shown in area A1 in Figure 4-44) are offered for selection in area A2 in Figure 4-44.

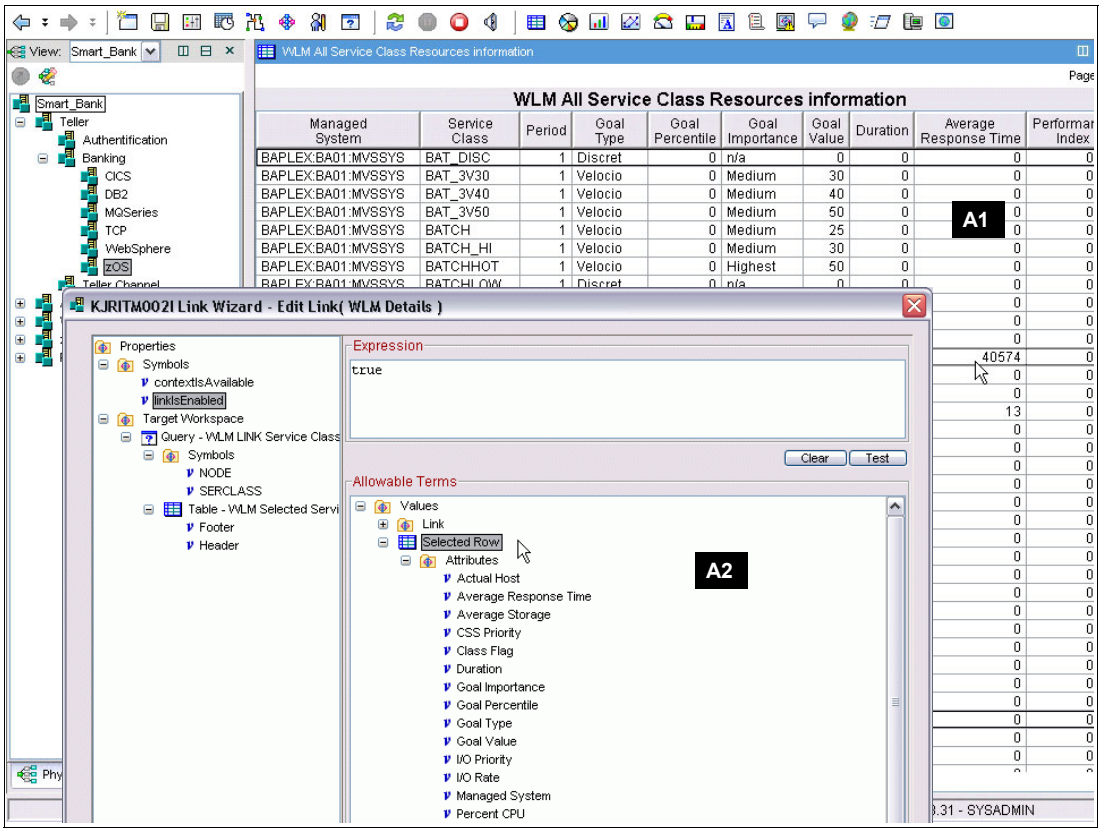


Figure 4-44 The attributes, as gathered by the attached query on the source workspace

Scroll down the advanced link definition window (zone A2 in Figure 4-44) to find the functions and operators you can use to build advanced filtering on the available attributes, as shown in Figure 4-45.

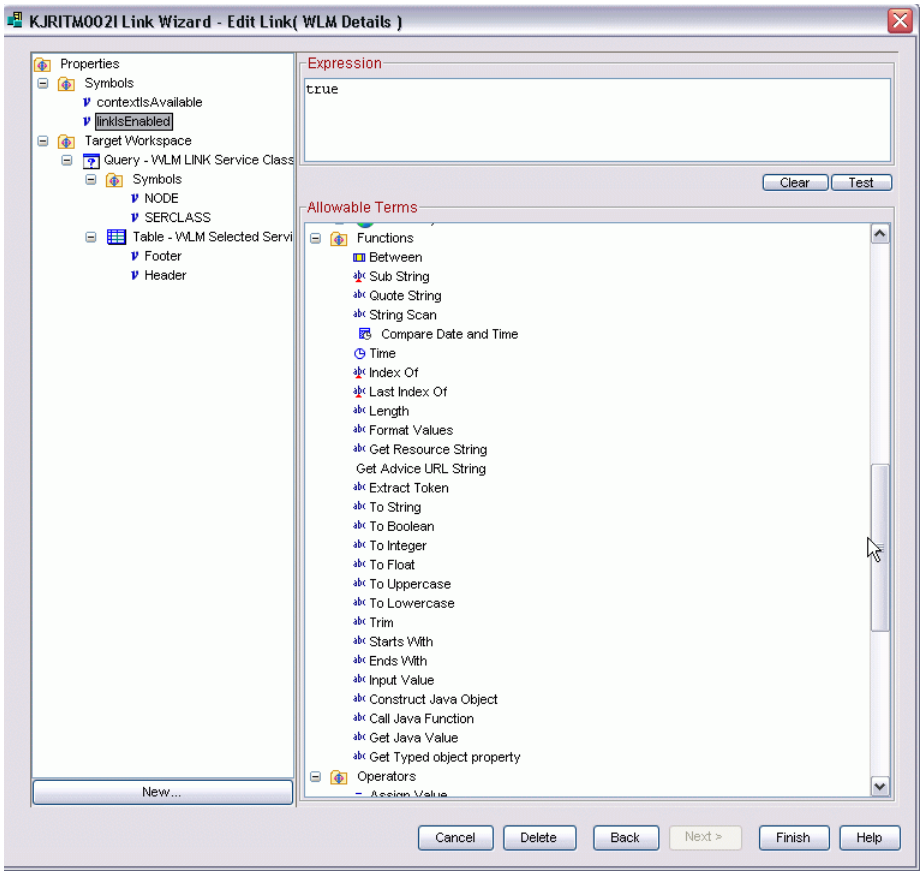


Figure 4-45 Dynamic link functions and operators

In summary, the advanced link definition window contains the following features:

- The structure of the target workspace
- Attributes gathered by the source workspace
- Functions to apply on them
- Operators to build some logic

Everything that is required to build the filter associations is now available.

Advanced link: Setting up a value for dynamic filtering

To set up a value for dynamic filtering in the advanced link, perform the following tasks:

1. In the right-hand side of the window (A in Figure 4-46), a Clear button is present. Click it to erase all the data in the top right section (B).
2. In the left side of the window, the symbol SERCLASS (C) is present. Click it.

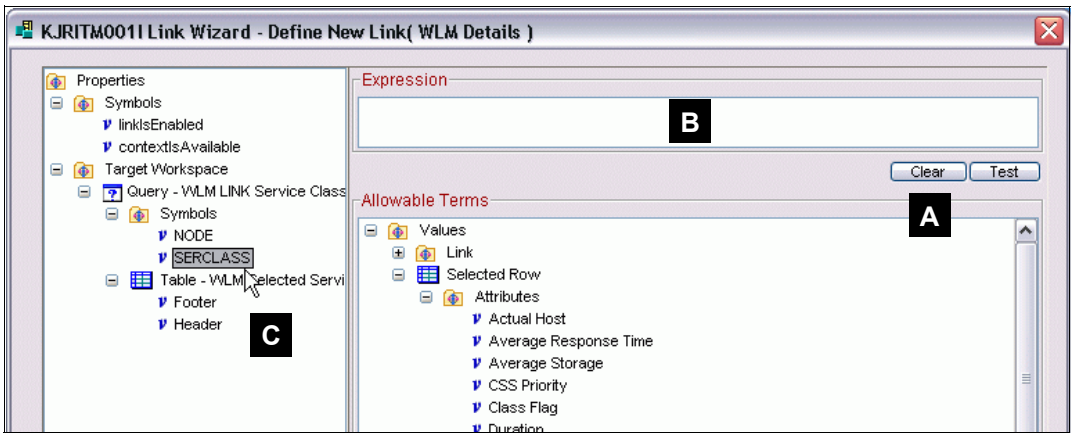


Figure 4-46 Selecting the SERCLASS symbol

3. In the bottom right section, in the Attributes list (zone A, inside Attributes, in Figure 4-47), click **Service Class** (B). This displays a string in the top right section (C).

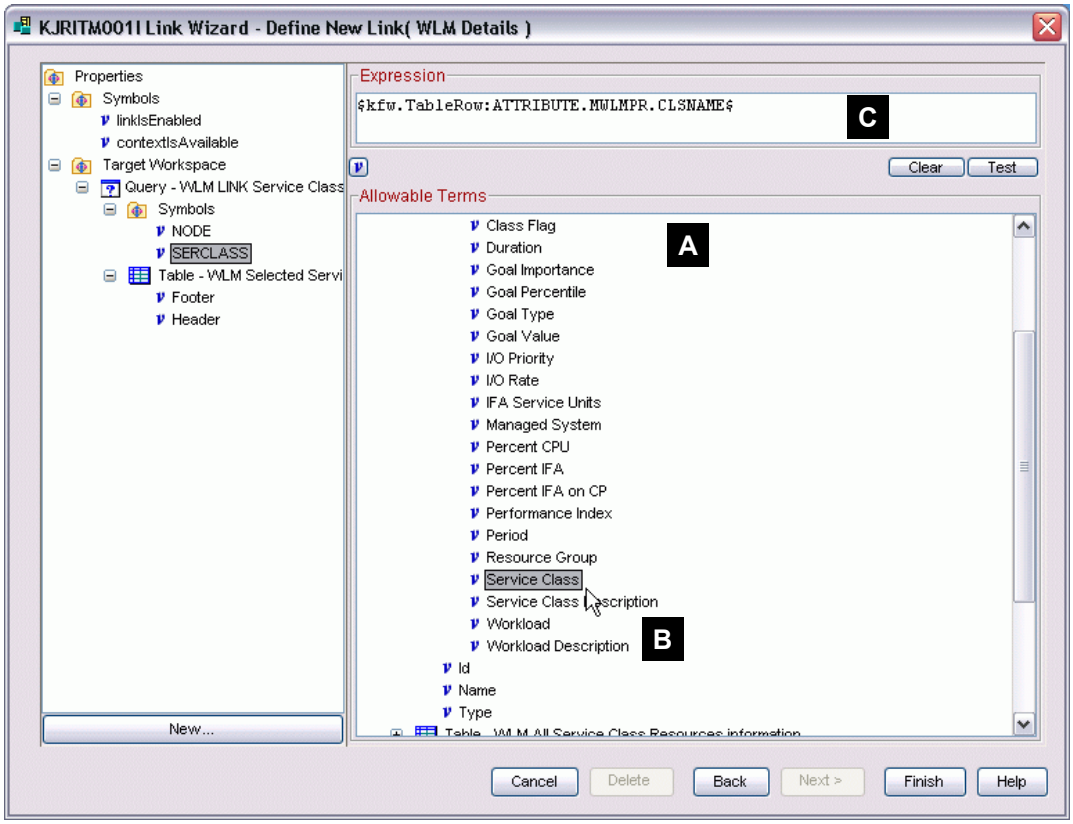


Figure 4-47 Selecting the Resource Class attributes

4. Click **Test** in zone A in Figure 4-48. A pop-up opens (B). It contains the name of the Service Class that is displayed in the line you selected (C) in the view at the beginning of the dynamic link definition.

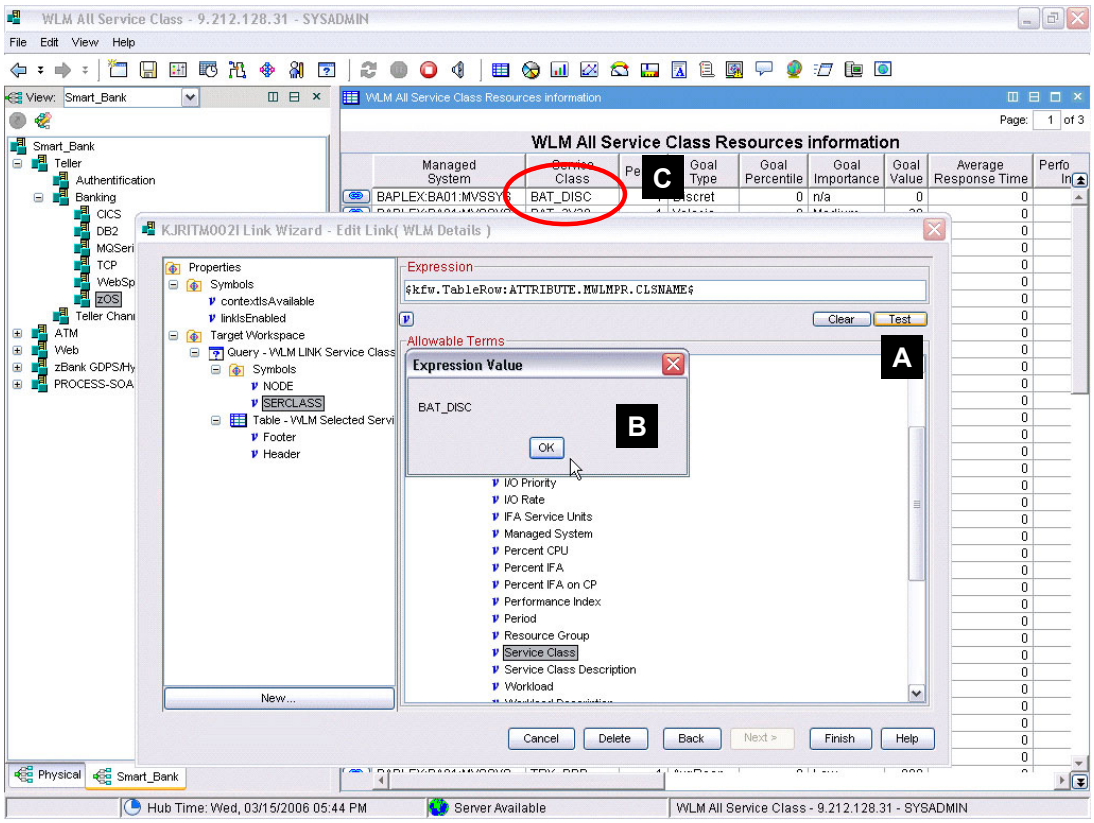


Figure 4-48 The result of the association

Tip: To fill the value of a symbol, you can also use a specific function called INPUT. This prompts the user for a value. OMEGAMON XE for MainFrame Networks uses this function in one of the views of the TELNET workspace, INPUT("Enter Remote IP", "Enter Value"). The first string (Enter Remote IP) is the entity to which the question is asked, and the second string (Enter Value) is the name of the pop-up. No verification is performed on the data.

5. A dynamic association between the SERCLASS symbol in the query and the value of the Service Class column in the line of the view is now realized. Click **Finish** to terminate this advanced link definition. The view named WLM All Service Class is now displayed (Figure 4-49) with a blue chain in the front, indicating that a link is available.

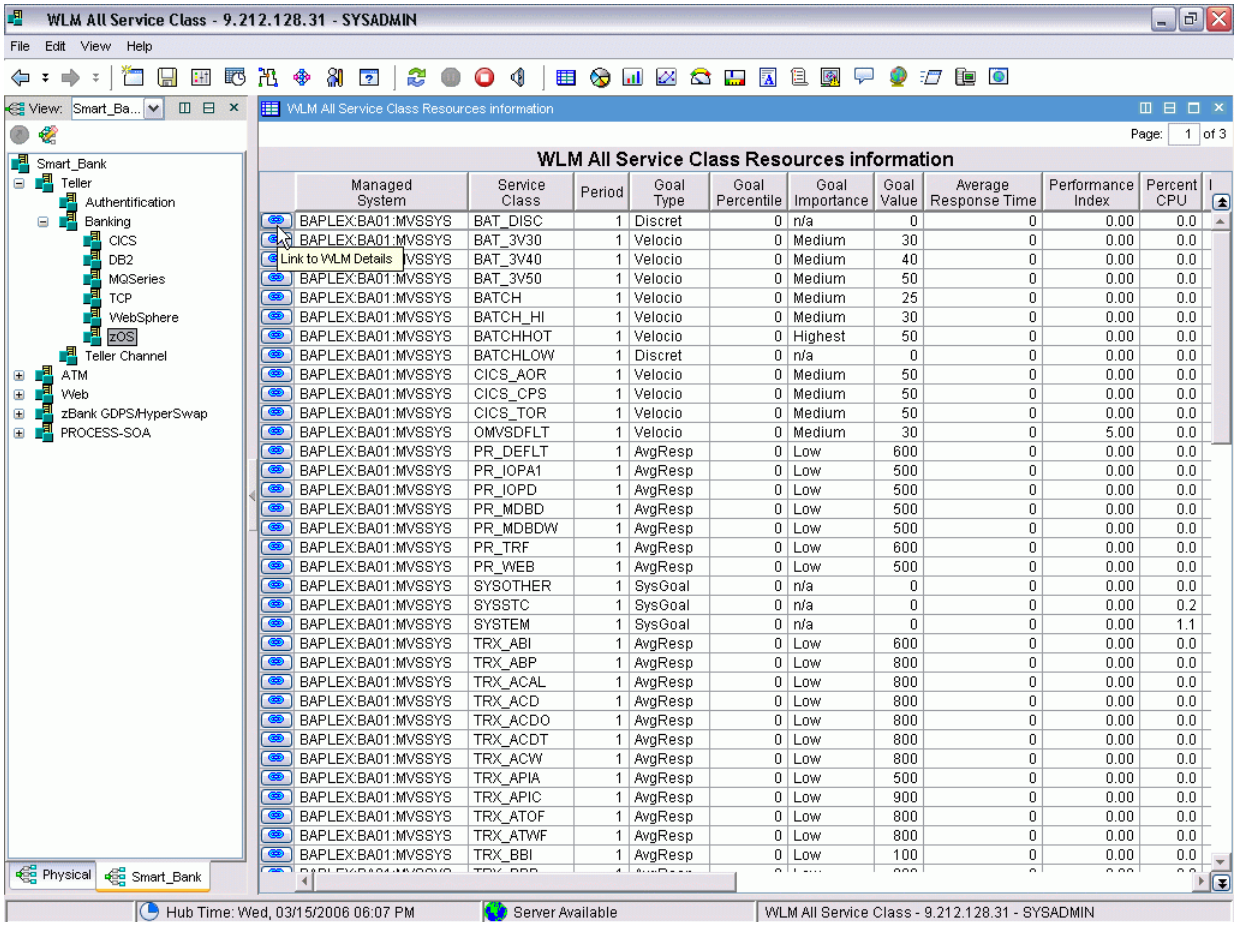

WLM All Service Class - 9.212.128.31 - SYSADMIN
File Edit View Help
View: Smart_Ba...
Smart_Bank
Teller
Authentification
Banking
CICS
DB2
MQSeries
TCP
WebSphere
zOS
Teller Channel
ATM
Web
zBank GDPS/HyperSwap
PROCESS-SOA
Physical
Smart_Bank
WLM All Service Class Resources information
Page: 1 of 3
WLM All Service Class Resources information
Link to WLM Details
Managed System | Service Class | Period | Goal Type | Goal Percentile | Goal Importance | Goal Value | Average Response Time | Performance Index | Percent CPU
BAPLEX:BA01:MVSSYS | BAT_DISC | 1 | Discret | 0 | n/a | 0 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | BAT_3V30 | 1 | Velocio | 0 | Medium | 30 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | BAT_3V40 | 1 | Velocio | 0 | Medium | 40 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | BAT_3V50 | 1 | Velocio | 0 | Medium | 50 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | BATCH | 1 | Velocio | 0 | Medium | 25 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | BATCH_HI | 1 | Velocio | 0 | Medium | 30 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | BATCHHOT | 1 | Velocio | 0 | Highest | 50 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | BATCHLOW | 1 | Discret | 0 | n/a | 0 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | CICS_AOR | 1 | Velocio | 0 | Medium | 50 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | CICS_CPS | 1 | Velocio | 0 | Medium | 50 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | CICS_TOR | 1 | Velocio | 0 | Medium | 50 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | OMVSDFLT | 1 | Velocio | 0 | Medium | 30 | 0 | 5.00 | 0.0
BAPLEX:BA01:MVSSYS | PR_DEFLT | 1 | AvgResp | 0 | Low | 600 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | PR_IOPA1 | 1 | AvgResp | 0 | Low | 500 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | PR_IOPD | 1 | AvgResp | 0 | Low | 500 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | PR_MDBD | 1 | AvgResp | 0 | Low | 500 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | PR_MDBDW | 1 | AvgResp | 0 | Low | 500 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | PR_TRF | 1 | AvgResp | 0 | Low | 600 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | PR_WEB | 1 | AvgResp | 0 | Low | 500 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | SYSOTHER | 1 | SysGoal | 0 | n/a | 0 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | SYSSTC | 1 | SysGoal | 0 | n/a | 0 | 0 | 0.00 | 0.2
BAPLEX:BA01:MVSSYS | SYSTEM | 1 | SysGoal | 0 | n/a | 0 | 0 | 0.00 | 1.1
BAPLEX:BA01:MVSSYS | TRX_ABI | 1 | AvgResp | 0 | Low | 600 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ABP | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ACAL | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ACD | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ACDO | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ACDT | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ACW | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_APIA | 1 | AvgResp | 0 | Low | 500 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_APIC | 1 | AvgResp | 0 | Low | 900 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ATOF | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_ATWF | 1 | AvgResp | 0 | Low | 800 | 0 | 0.00 | 0.0
BAPLEX:BA01:MVSSYS | TRX_BBI | 1 | AvgResp | 0 | Low | 100 | 0 | 0.00 | 0.0
Hub Time: Wed, 03/15/2006 06:07 PM
Server Available
WLM All Service Class - 9.212.128.31 - SYSADMIN

Figure 4-49 The link on the WLM All Service Class view

6. If you click the link, it selects the value in the Service Class column. It uses it as a filter by setting the SERCLASS symbol to this value. It then displays the filtered target workspace

(Figure 4-50) with only two lines, one for each of the Smart Bank showcase image (BA01 and BA02).

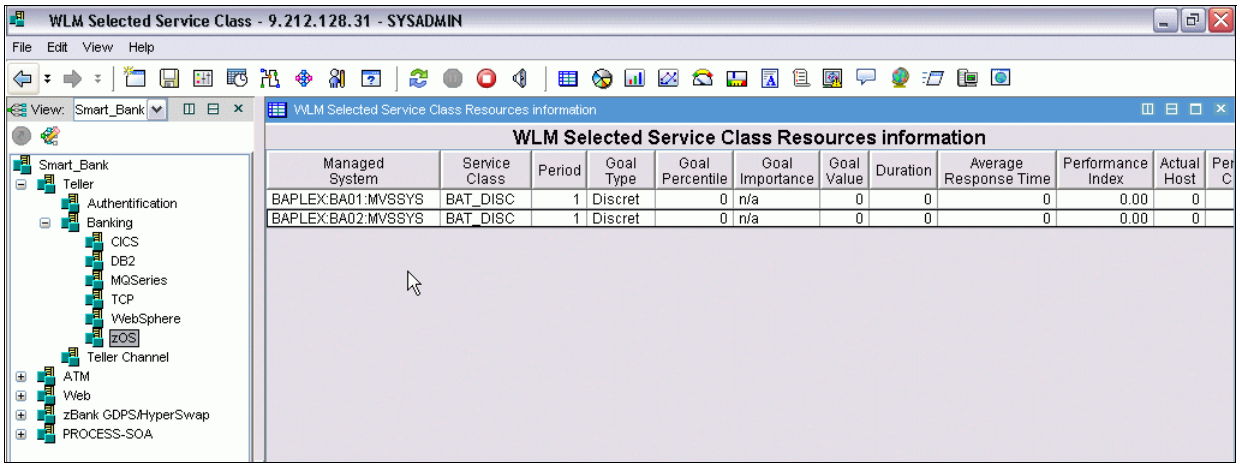


Figure 4-50 Result of the dynamic link utilization

Advanced link: Going further, the header and the footer

With the advanced link that is defined, you move from a view called WM All Service Class Resources information to another view called WLM Selected Service Class Resources information. It is better to present the target view with a title that includes, as a sample, the selected Service Class and its description. You can perform this task in the advanced link definition.

Note: Set up the text of a header or a footer the same way that you assigned a value to a symbol in a query. Click the header or footer you want to set up (in the left-hand side of the window), and an attribute in the right-hand side of the window. By doing so, the original text, if any, of the header or the footer is disregarded. The *title* of the view is changed to the new assigned value. If you do not select the Split Pane Toolbar option in the TEP view menu, this title is not displayed.

Advanced link: Inserting symbols for the header and the footer

In the advanced link definition window, new symbols can be added. They work in the same way as the symbols used through queries.

To create a new symbol, perform the following tasks:

1. Click **New** at the bottom left of the advanced link definition window (zone A in Figure 4-51).

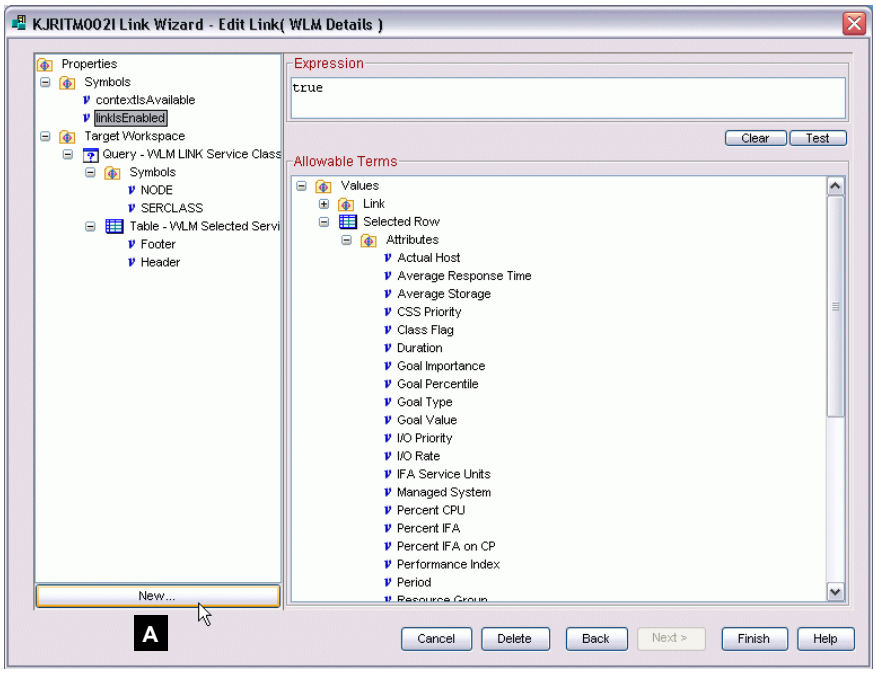


Figure 4-51 Adding a new symbol for a dynamic link definition

2. A pop-up opens. Enter the name of the new symbol, as shown in Figure 4-52. In our case, the name ProcHeader is used.

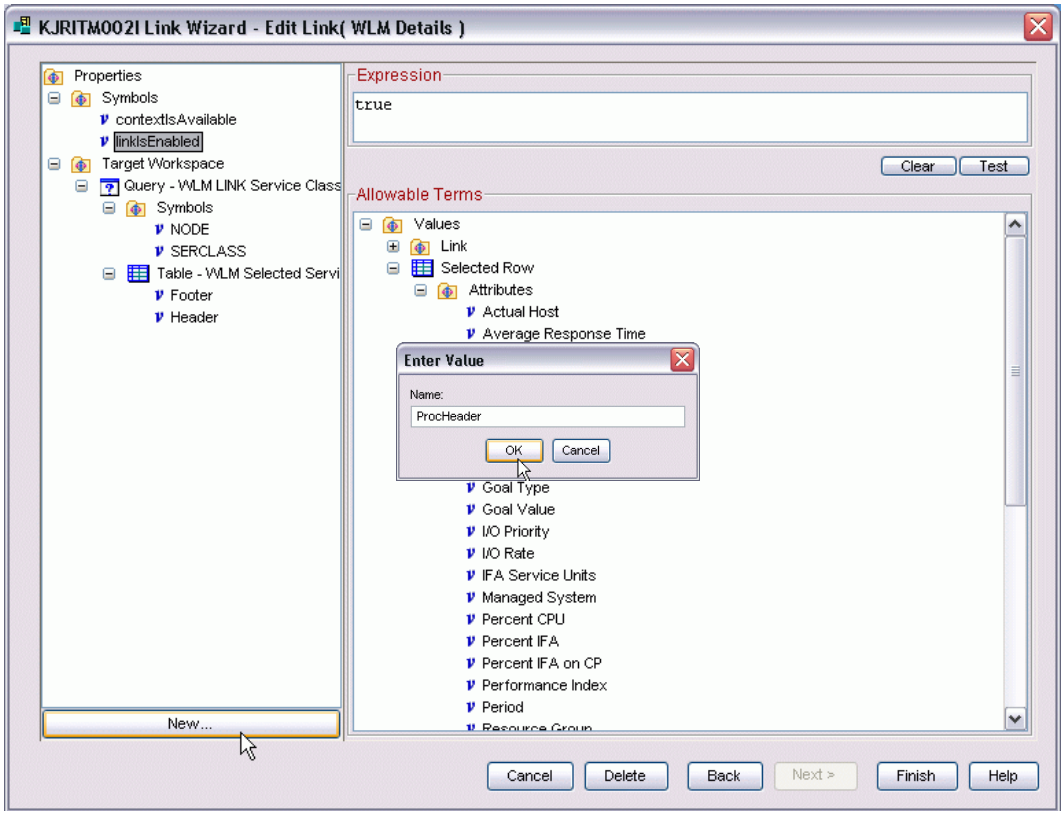


Figure 4-52 Naming the new symbol

3. The newly defined symbol is then placed in the left-hand side of the window (A in Figure 4-53). As with the symbol in a query, select an attribute (B) in the bottom right section in order to assign it a value (C).

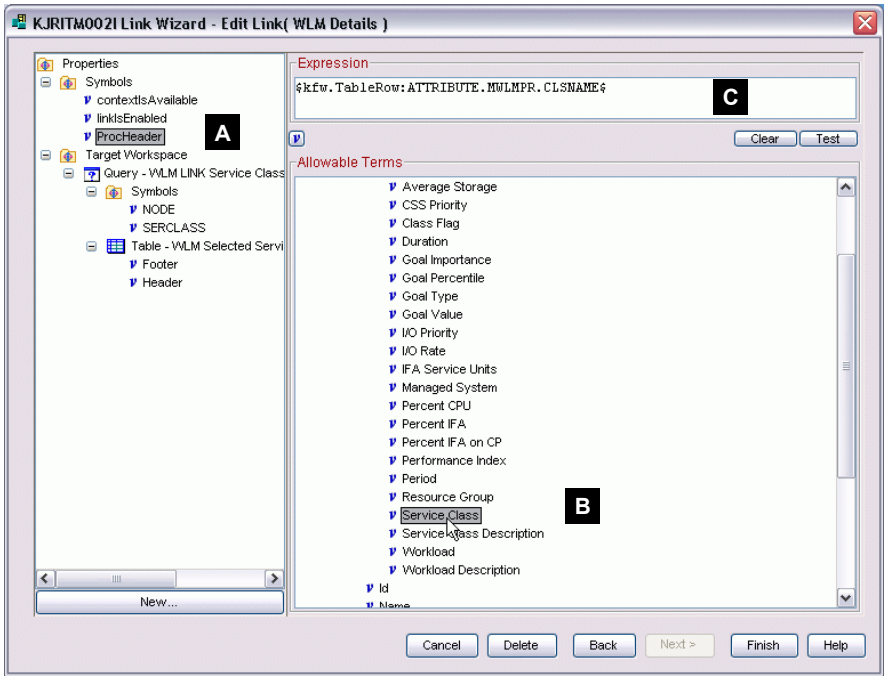


Figure 4-53 Assigning value to the new symbol

Advanced link: Assigning value using operators and functions

Set up a symbol's value by using one of the following:

- Attributes setup
- Strings
- Functions applied on attributes
- Operators

1. To complete the value of the current symbol, add a string such as *selected*. In our case, the operator plus (+) is used to concatenate this string with the current value (Figure 4-54). Open the function tree in the bottom right section of the definition window. Select the **TOLOWER** function (conversion of a string to lower case).

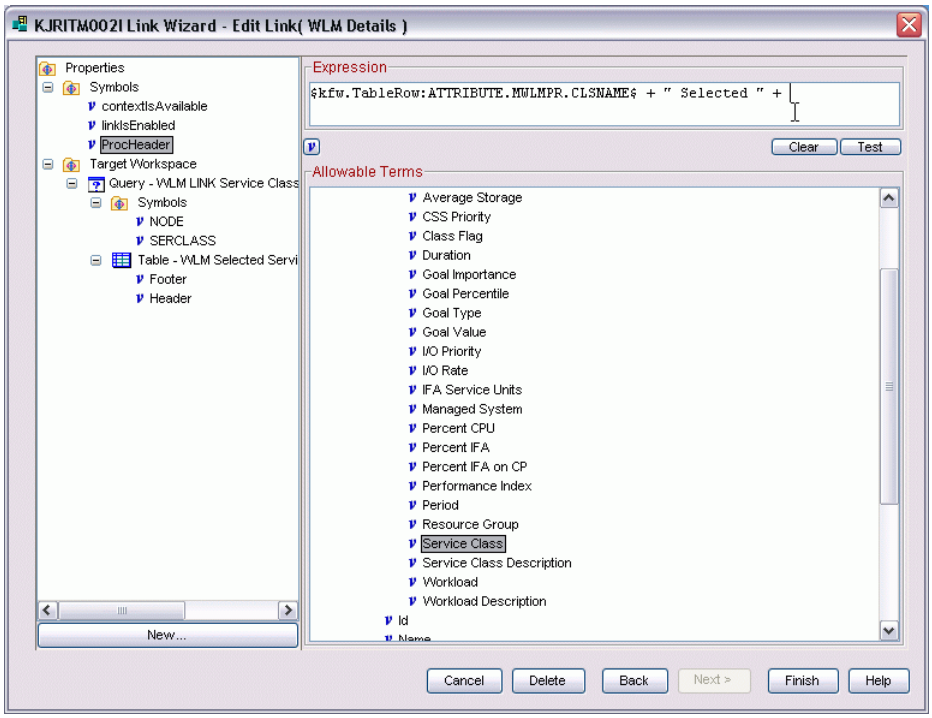


Figure 4-54 Completing the value of the symbol

2. The skeleton of this function is shown (Figure 4-55).

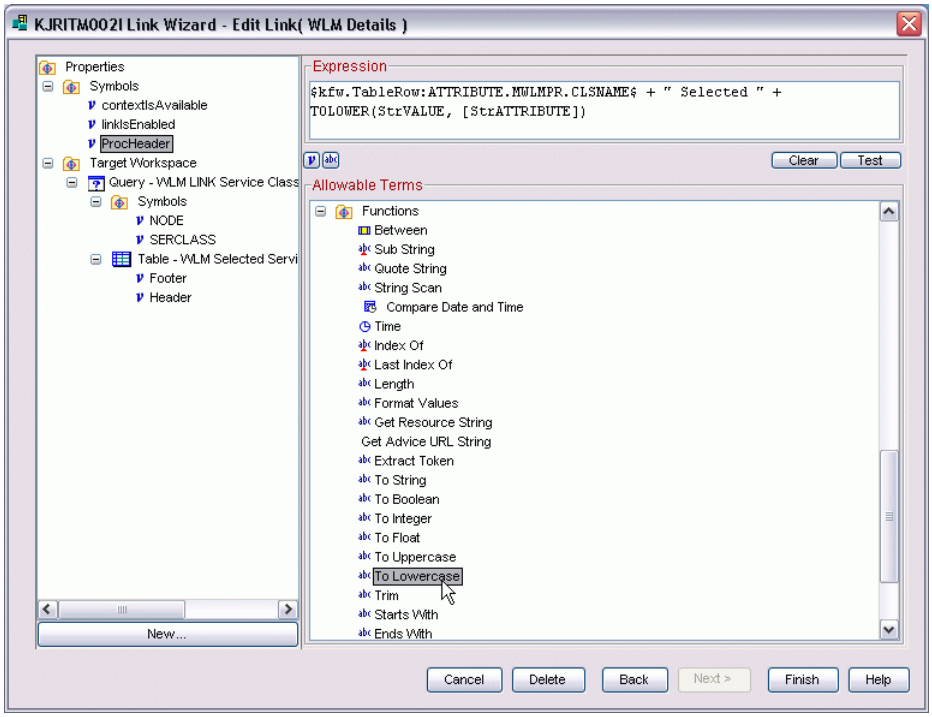


Figure 4-55 Using a function to set up a symbol value

3. To set up the function parameters, erase the proposed default (Figure 4-56).

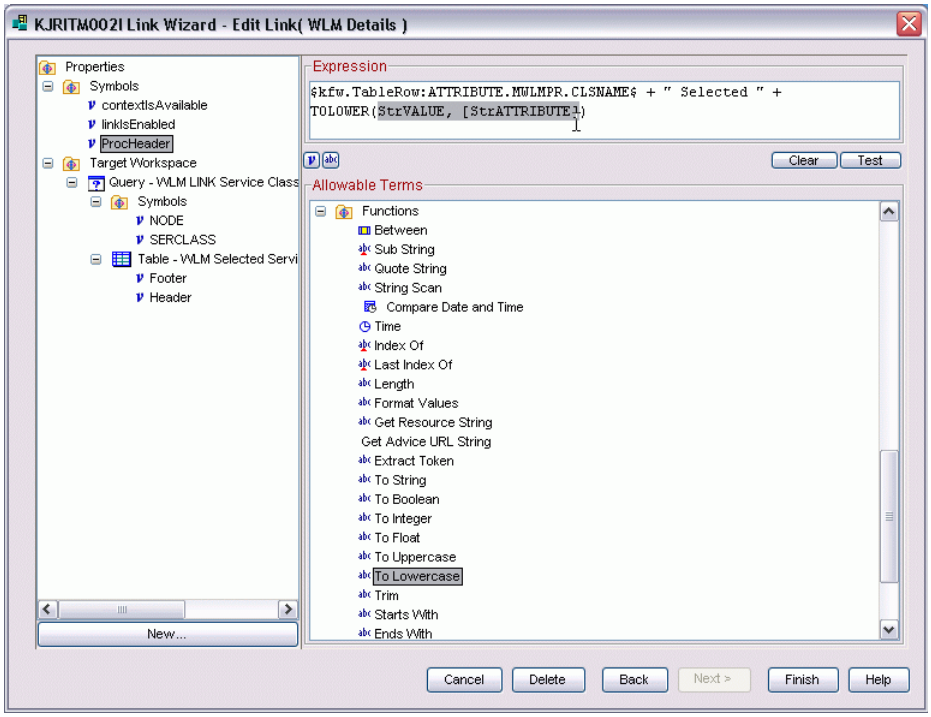


Figure 4-56 Modifying the function parameter

4. To fill the required parameters for this function, in the attribute list, select the one containing the Service Class description. The top right section of the dynamic link definition is updated (Figure 4-57).

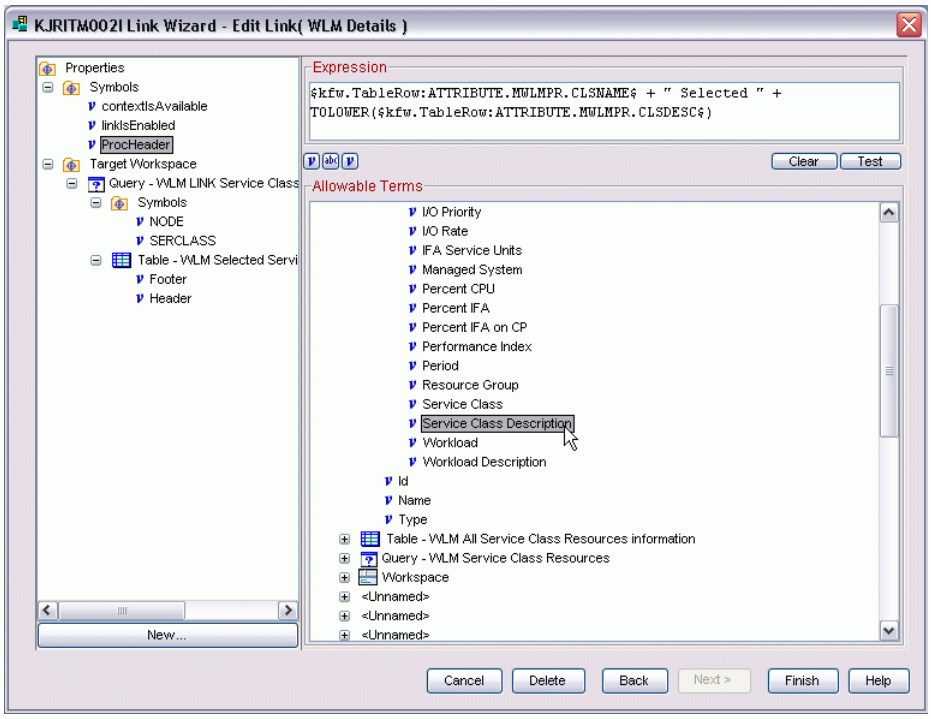


Figure 4-57 Attribute as a value of the function parameter

5. A new symbol is ready. Its value is a concatenation of the attribute that contains the name of the service class. Select a string. Then, change the contents of the Service Class Description attribute to lower case. Click **Test** to ask the TEP to validate the formula (zone A in Figure 4-58). If there is no error, the result of the expression is displayed in a pop-up (B).

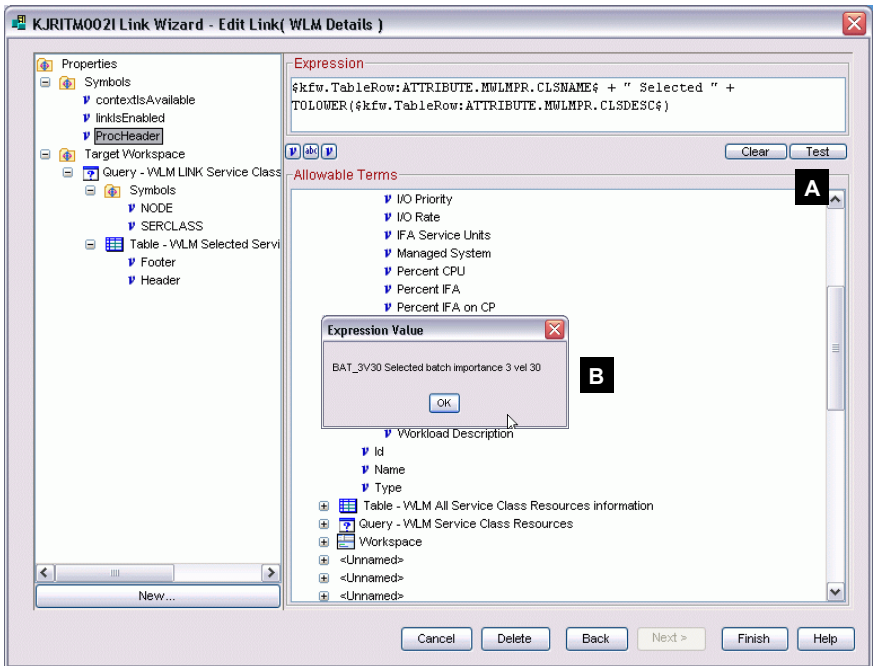


Figure 4-58 Validation of the formula

A new symbol is defined along with an expression to determine its value. A part of this value is filled with some of the attributes coming from the source view.

Tip: You can use the same process (functions and operators) to fill the value of the symbols defined in any of the target workspace queries.

Advanced link: Full usage of a newly defined symbol

The target window of the dynamic link is the workspace named WLM Selected Service Class. Open that workspace, and perform the following tasks:

1. Select the table view and open **Properties**.
2. In the Style section of properties, select **Header**.
3. Check the box against **Show**.
4. In the Text field, enter the name of the symbol between the two dollar ($$) signs (Figure 4-59).

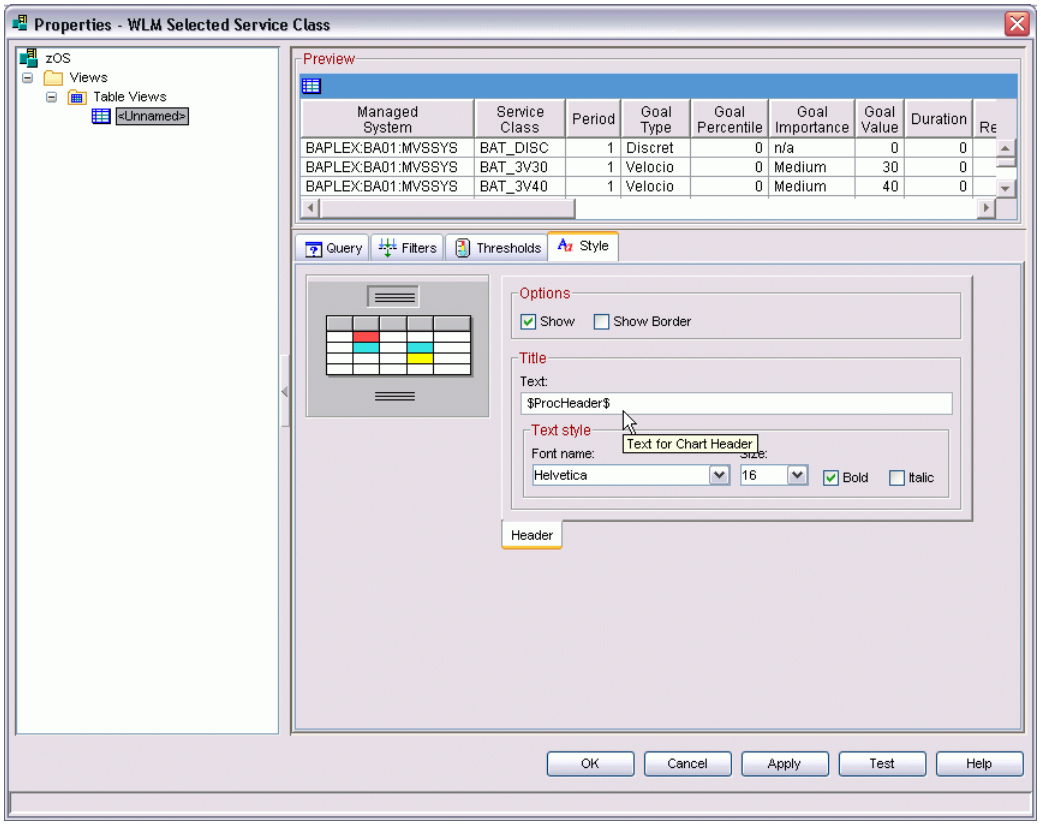


Figure 4-59 Using the new symbol

Advanced link: The full sample

Just as a user will, open the **WLM All Service Class** workspace (Figure 4-49 on page 230). When you click the blue chain, the target workspace displays the now filtered data *and* a new header that is the substitution of the ProcHeader symbol as a result of the expression. A dynamic link can not only filter the data, but also dynamically adapt the contents of the view (Figure 4-60).

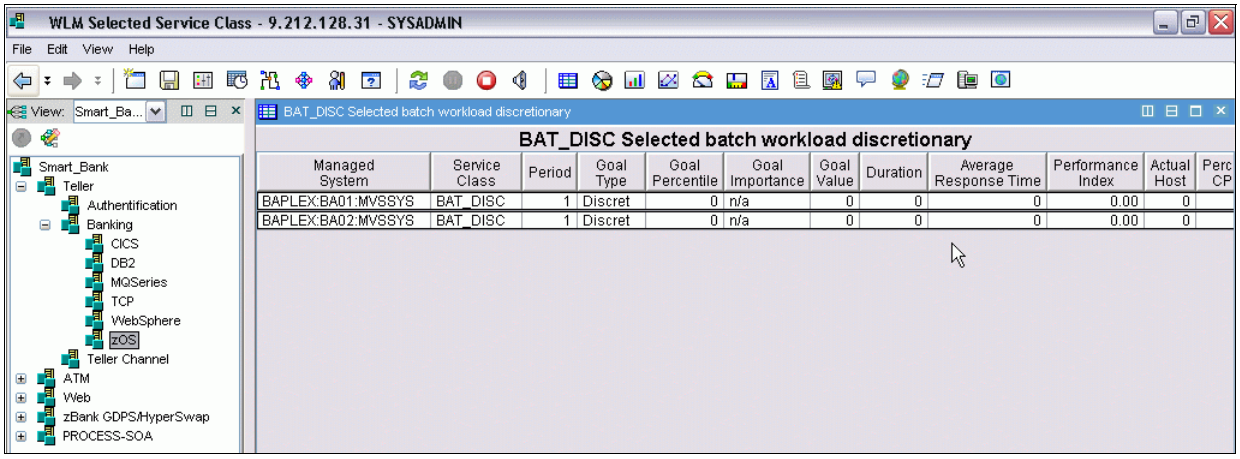


Figure 4-60 Dynamic link: The full sample

An advanced link definition can do much more. You can build headers, footers, and the value of the symbol used in queries. Thus, parts of the attributes' values can be concatenated to build a new value that will be assigned to a symbol in a query.

Advanced link: Remarks

This section describes some, but not all the advanced link features. By referring to guides and other relevant documentation, you can find more information about all the functions and how to use them. You will find that in headers and footers, you can mix the string values and symbol values, and that, when in the dynamic link definition window, you can click some of the displayed fields in order to find out the variables that can be used in the definition processes.

The dynamic link is a powerful tool in which you will discover something new every time.

Note: In the same view, you can define as many links as you want. If you do so, the user will see all the defined links. The default link will be presented first, but the user can choose another one. If the user clicks the link icon directly, the default link is used.

Link anchor

In the link anchor window, you can perform the following tasks:

- Select the default link if more than one link is defined
- Decide whether to display or not display the link anchor (the blue chain)
- Offer conditional link only on the lines in a table view or icons in a graphic view, which match the conditions (the value of linkIsEnabled in the dynamic link definition window)

In the same target workspace used earlier, add a new link. Open the link anchor window to set up the default options and the other options (Figure 4-61). In this case, **WLM details** is selected as the default link. To see the blue chain icon and show this link on every line, check the boxes against **Show Link Indicator** and **Link Indicator Always Enabled** in the pop-up.

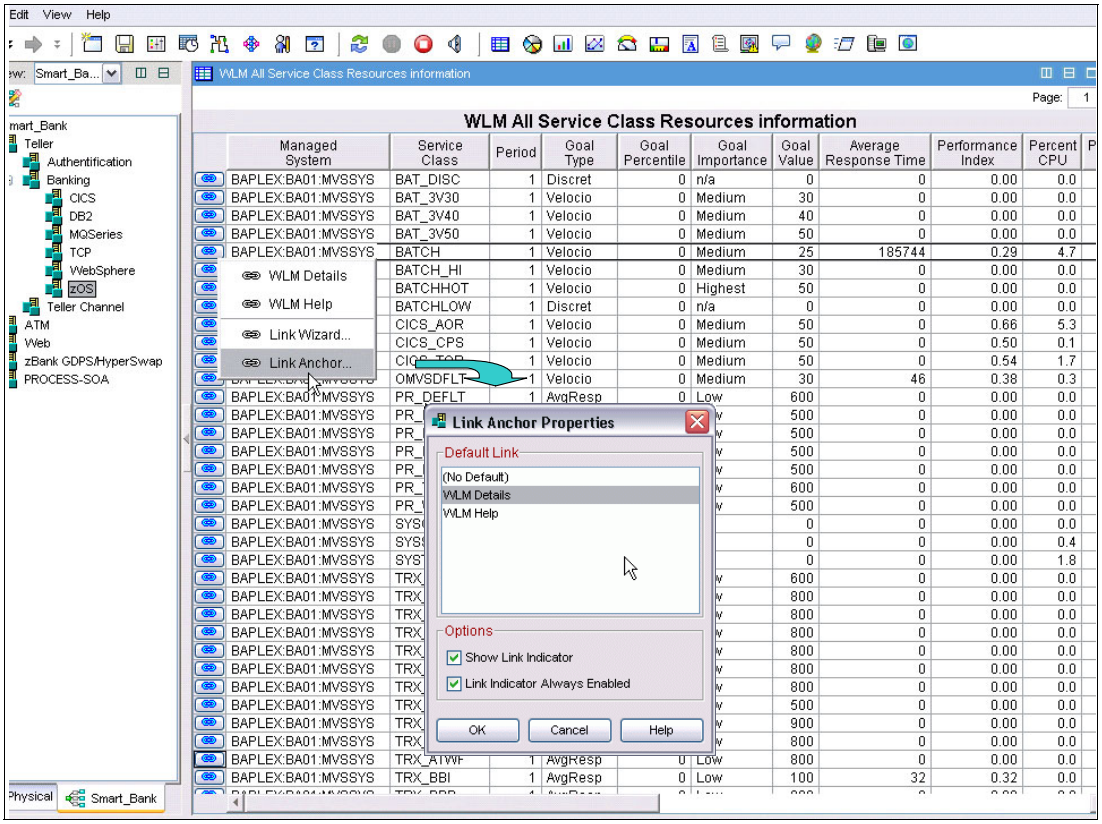


Figure 4-61 The link anchor window

4.5 Style in the Logical view

A Logical view is a way of organizing data according to the requirements of the user. The style features allow you to personalize the Logical views.

4.5.1 Using the standard Tivoli Enterprise Portal

The Smart Bank showcase team designed a picture that they used for advertising. This is in T:\IBM\ITM\CNB\classes\candle\fw\resources\backgrounds\user, which is the TEP directory dedicated to the user background (png, jpg, or gif files).

When used as a background in the workspace located at the root level of the Smart Bank showcase Logical view, it opens as shown in Figure 4-62.

Figure 4-62 Smart Bank showcase: First level

4.5.2 Using the Smart Bank showcase style

Using the style features, the Smart Bank team designed an alternative to the previous display (Figure 4-62). The first level workspace provides access to the data in another way, as shown in Figure 4-63. The standard icons have been changed and replaced by pictures.

Figure 4-63 The Smart Bank showcase style

4.5.3 Setting up a style in a graphic view

To set up a style in a graphic view, perform the following tasks:

1. When the graphic view is displayed, right-click it to access its property menu (Figure 4-64). Click the plot area to access the background and style setup.

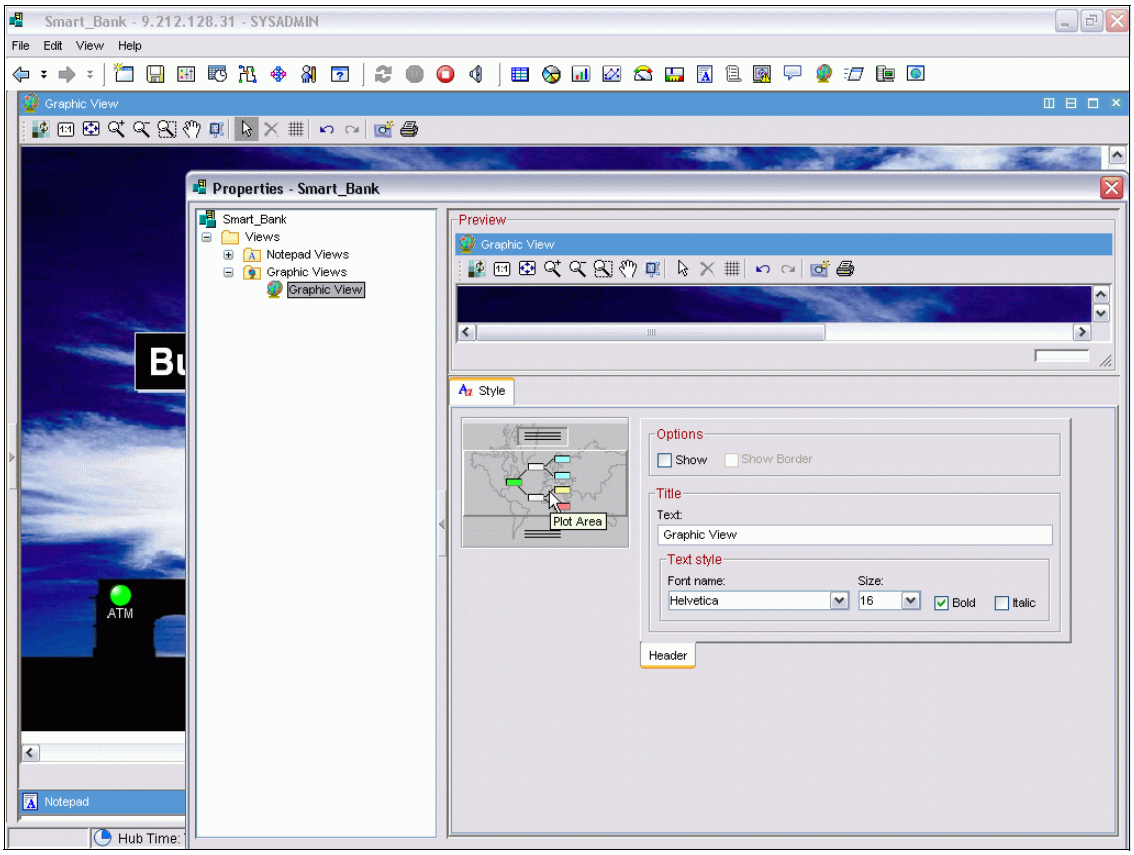


Figure 4-64 Graphics property access

2. To change the current style to the Smart Bank showcase style, click **Browse** (Figure 4-65).

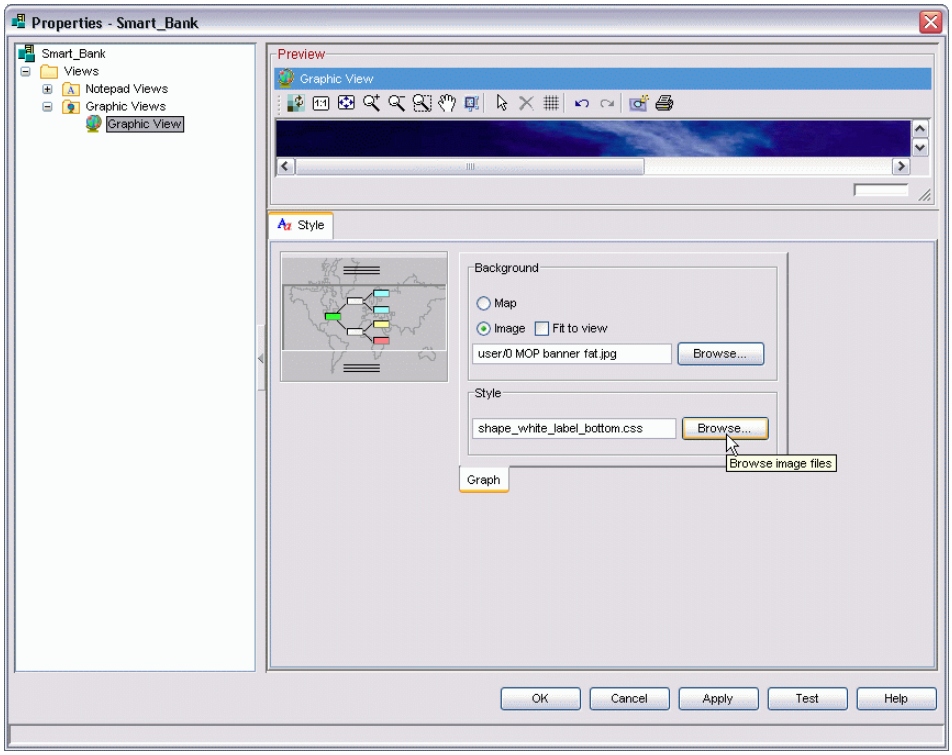


Figure 4-65 Background and style setup

3. Open the User folder and select the entry (created earlier) named **Smart Bank showcase.css** and open it (Figure 4-66).

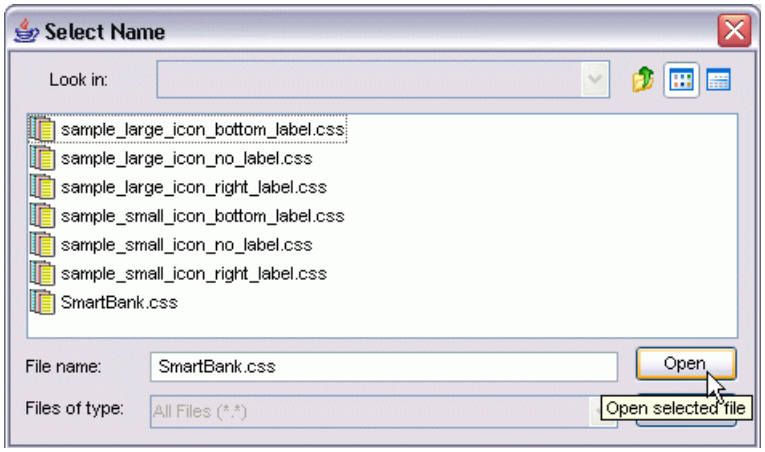


Figure 4-66 Opening Smart Bank showcase style

4. The style window closes. Click **OK** in the background and style setup window. Return to the graphic display. The icons have changed and have been replaced by images, as shown in Figure 4-63 on page 242.

4.5.4 The Smart Bank showcase.css style file

The TEP uses css files. This section focuses on the use of these files by the Smart Bank showcase team. In order to build its own style, the Smart Bank showcase team first gathered some icons (gif and jpg). The images were then modified to fit the following parameters:

- Large icons created as close to 32 x 32 pixels as possible
- Small icons created as 20 x 20 pixels
- GIFs may have transparent backgrounds and may be animated

An image that is larger or smaller is resized to these dimensions automatically and may look distorted. Also, the image retains the proper proportions only if it is of the same height and width.

In the directory named T:\IBM\ITM\CNB\classes\candle\fw\resources\style\user, a file named Smart Bank showcase.css is created using Windows Notepad. The purpose of the css file is to perform the following functions:

- Define the base style
- Set up large or small icons
- Place the label
- Set up the color of the label
- Assign an image to each node name used

The content of the Smart Bank showcase.css file is shown in Example 4-3.

Example 4-3 The Smart Bank showcase.css style file

```
@import "../$base.css";
@import "../$baseLargeIcon.css";
@import "../$baseNodeLabelBottom.css";

node {
        foreground : "white";
}
node[name = "ATM"]              { iconImage : url(../../icons/user/ATM32.jpg); }
node[name = "Teller"]              { iconImage :
url(../../icons/user/TELLER32.jpg); }
node[name = "PROCESS-SOA"]      { iconImage : url(../../icons/user/SOA.gif); }
node[name = "Web"]              { iconImage : url(../../icons/user/WEB32.gif); }
node[name = "zBank GDPS/HyperSwap"]     { iconImage :
url(../../icons/user/DISK32.jpg); }
```

For each node included in the graphic view, the Smart Bank showcase.css file associates an *iconImage*. The corresponding image must be included in the defined directory.

Each node name given in the Smart Bank showcase.css file *must exactly* match the one given in the Logical view. If not, the default style is used instead.

You can play with the css contents files using the numerous samples provided. As an example, modify the foreground color in the Smart Bank showcase.css file from white to black. The icons legend in the Smart Bank root graphic is displayed in black. Example 4-4 shows how to suppress the legend of the icons.

Example 4-4 Suppressing the legend of the icons

```
@import "../$base.css";
@import "../$baseLargeIcon.css";

node {Anchor           : "Center";
      labelPosition    : "None"; }

node[name = "Ming's Kitchen"]     { iconImage :
url(../../icons/user/ming_logo.gif); }
node[name = "Human Resources"]    { iconImage :
url(../../icons/user/human_resources.gif); }
node[name = "Merchandising"]      { iconImage :
url(../../icons/user/merchandising.gif); }
node[name = "Retail"]             { iconImage : url(../../icons/user/retail.gif);
}
node[name = "Shipping"]           { iconImage :
url(../../icons/user/shipping.gif); }
```

Note: A style is applied on one view. You can use different styles for each graphic view on a single workspace.

4.5.5 The example: The Smart Bank showcase Logical view

The objective of the Smart Bank showcase is to demonstrate the value of IBM infrastructure in the context of retail banking. The Logical view designed by the Smart Bank showcase team must support this objective. As a consequence, this Logical view does not show the data that demonstrates the health of the systems, but the data that is the backbone of the showcase. In fact, it is a Logical view designed to support the application the showcase represents. This section introduces the organization of the Smart Bank showcase Logical view that is built to support this objective.

The Smart Bank main menu

At the time of logging in, the user receives a welcome menu, as shown in Figure 4-67. This menu introduces the showcase.

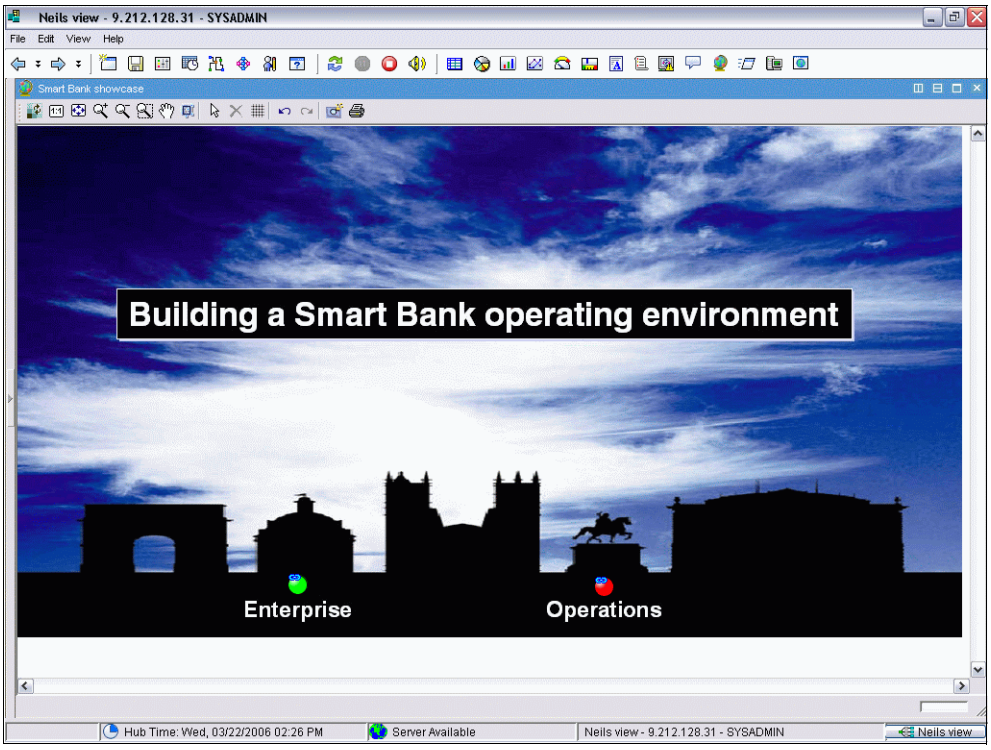


Figure 4-67 The Smart Bank welcome menu

The user has the following choices:

- To examine the enterprise, which is essentially the services that clients actually receive
- To have direct access to the components of the Smart Bank infrastructure

The blue chains that surround the two round icons indicate to the users that they can click it to go to another display or workspace. In this case, the Smart Bank team set up two absolute links, one to go to the Enterprise view and another to go to the Operations view.

Note: The Smart Bank showcase Logical view does *not* use many dynamic links because there are few requirements to extract data from one view to filter on another workspace during the showcase.

The Smart Bank Logical view tree

The structure of the complete Logical view is shown in Figure 4-68.

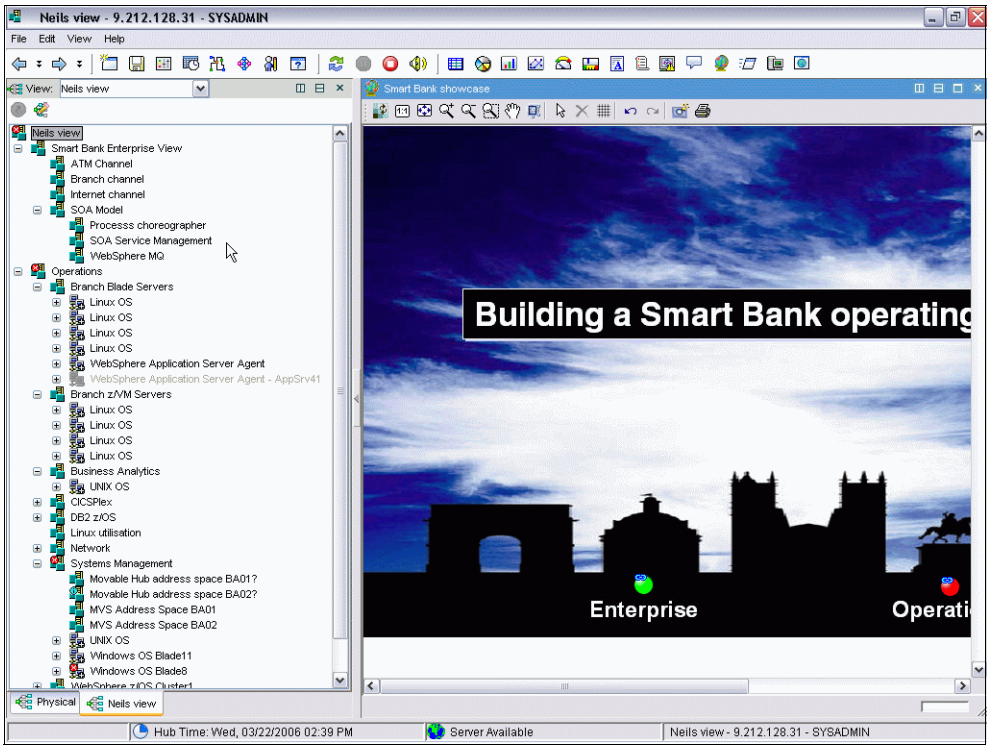


Figure 4-68 The Smart Bank Logical view tree

The Logical view tree is clearly defined in two parts:

- The first one is the structure of the Smart Bank Enterprise view. It is simple because we want to display the services the Smart Bank clients receive through the different channels offered.
- The second one is more detailed. The Smart Bank team picked up all the physical environments involved for each channel. The categories and subcategories are organized according to the decisions taken by the Smart Bank showcase team. The main idea is to create groups of identical involved hardware, middleware, and software components. These subtrees are designed at the time of creating new children, or are simply created by copying the adequate parts of the physical tree directly. New situations are defined on some levels to complete the set of situations inherited from the copies of the physical tree.

As an example, a specific situation is defined in the Moveable Hub address space BA01 level (Figure 4-69).

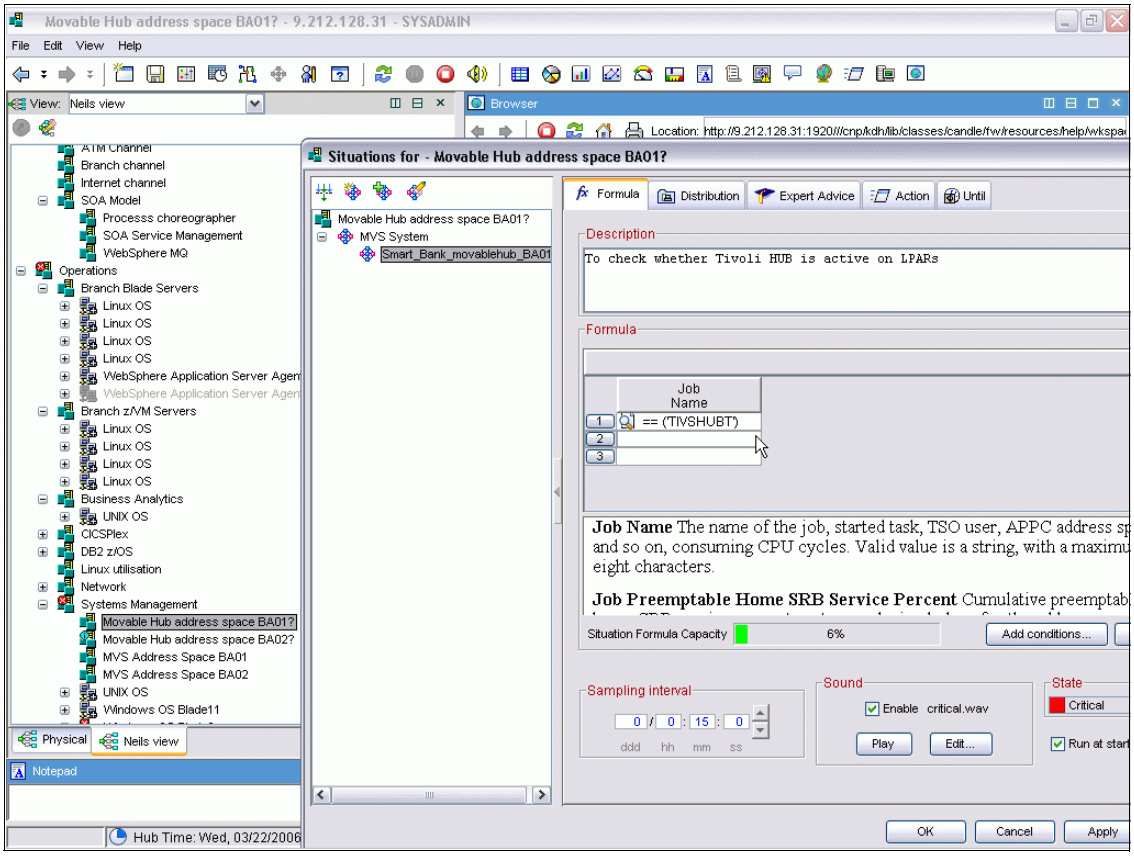


Figure 4-69 A new situation at the Moveable Hub level

Using the Smart Bank Logical view

When you click the Operations link in the Smart Bank showcase welcome menu (Figure 4-67 on page 247), a new workspace, the Smart Bank showcase Operations workspace (Figure 4-70), is displayed.

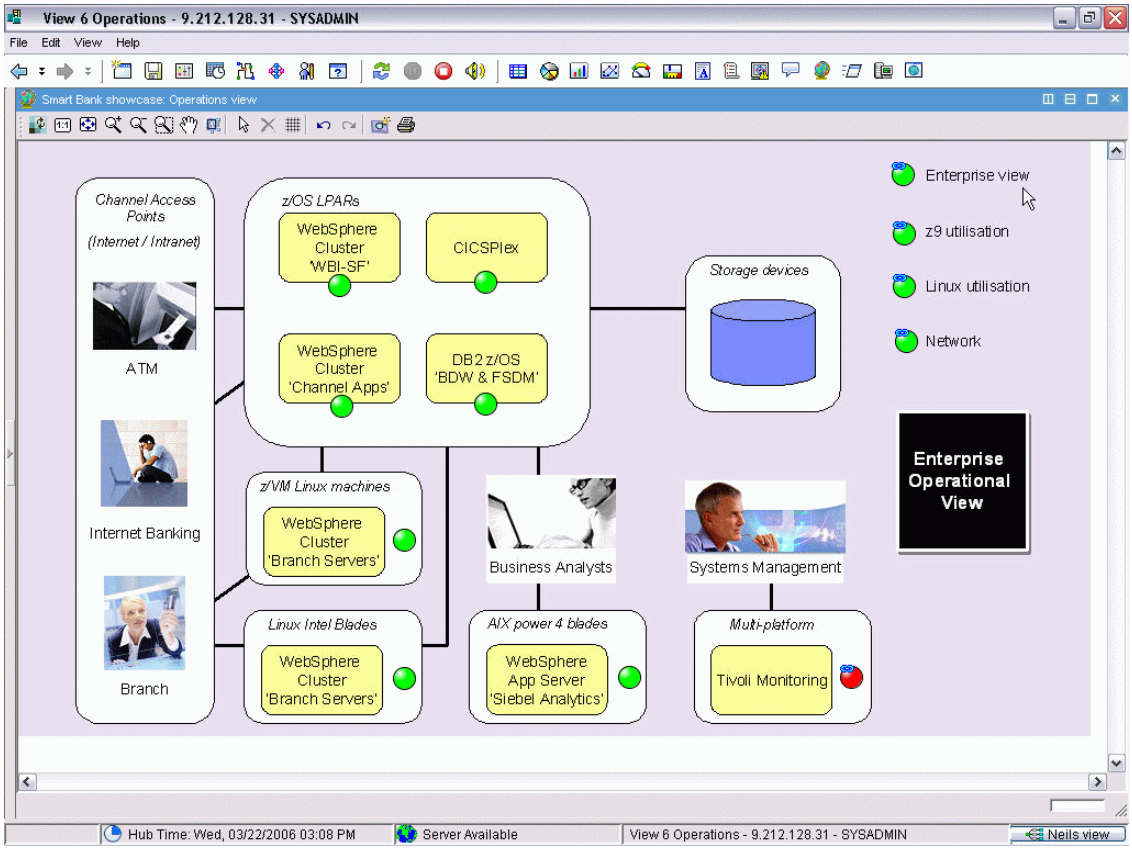


Figure 4-70 The Smart Bank operations workspace

This workspace has one view, a graphic one. Following are some of the features of this workspace:

- An enterprise link is offered in this view

  A user has the choice to go to the Enterprise view from this point. The user does not have to return back.

- Extra choices are added

  This includes, for example, an examination of the network or the IBM System z usage, and so on, that is, much more physical data pertaining to the Smart Bank application components.

- Icons with “nolabel” (corresponding style used) on the graphic

  These labels are integrated as part of the graphic.

- An extra part called Systems Management

  This is information pertaining to how the different Tivoli components actually measure the infrastructure.

Adding a workspace that displays the status of the different supervision products is a good idea. At any time, the Smart Bank team can see if any of the agents is falling, or if one of the platforms on which a Tivoli key component resides, is facing problems. The Enterprise status

and the Tivoli monitoring status are displayed on only one terminal (Figure 4-71, where the Windows station on which the Tivoli Enterprise Portal Server resides, is facing a problem).

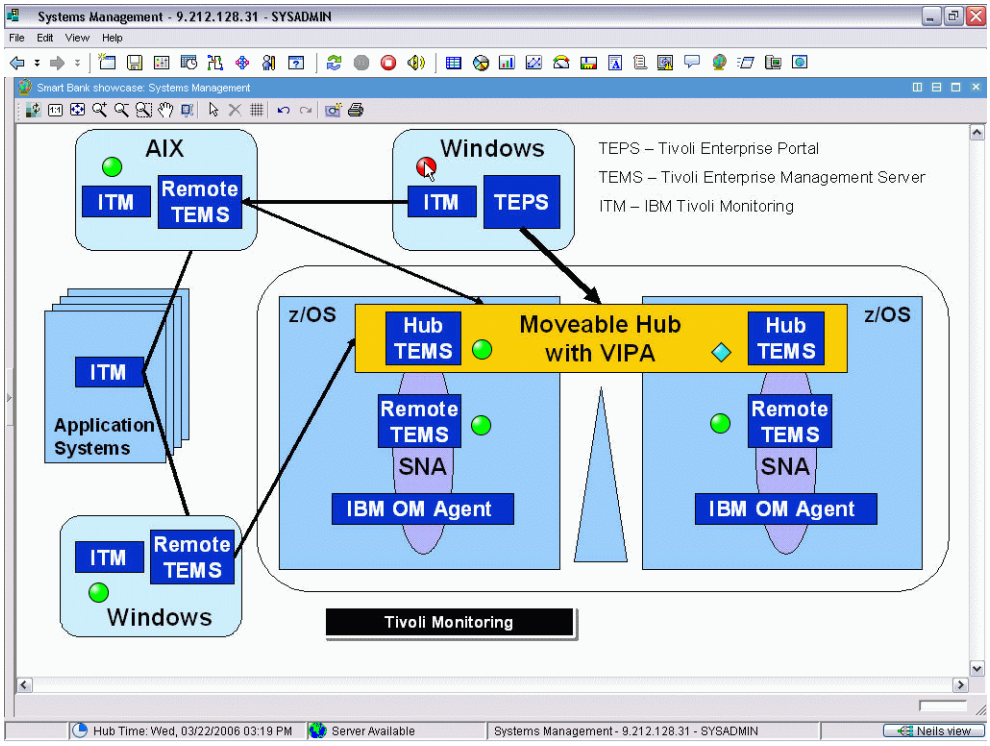


Figure 4-71 The Tivoli Systems Management workspace

If, from the Operations workspace (Figure 4-70), a user chooses the Network link, a new workspace opens (Figure 4-72).

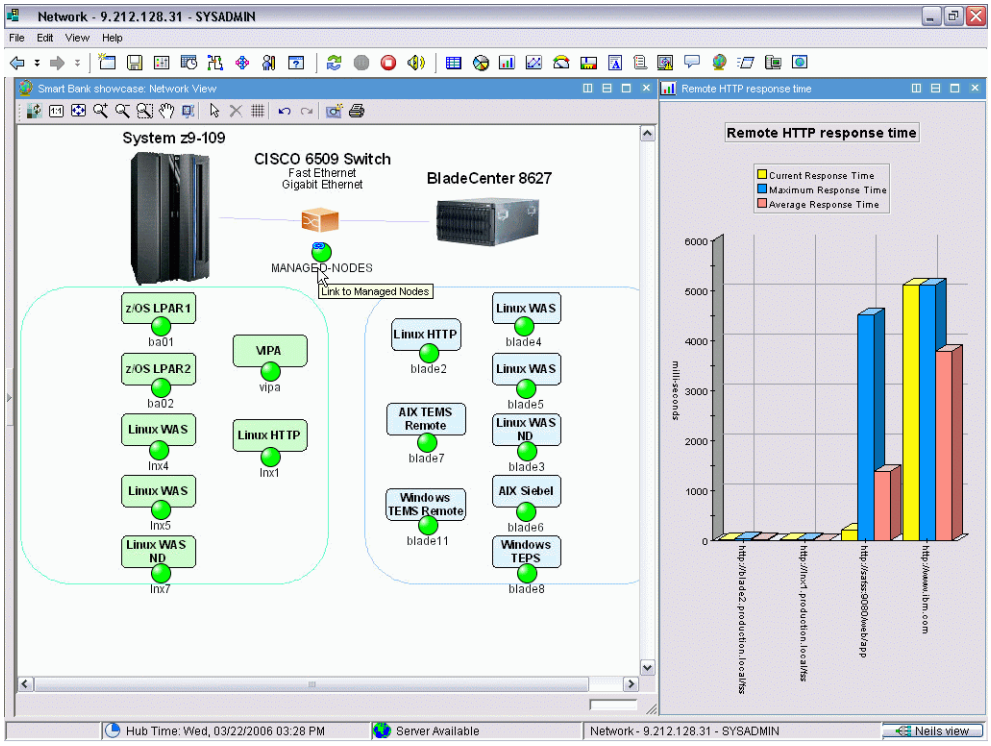


Figure 4-72 The Smart Bank Network workspace

Following are some features pertaining to this workspace:

- Most of the Smart Bank showcase workspaces contain graphics. They are easy to read, can offer new links, and are easy to use
- Care must be taken while designing a graphic, preferably built by using Microsoft PowerPoint®
- Pictures are often included in the graphics

Resuming the Smart Bank Logical view

The Smart Bank showcase Logical view is designed to support the showcase. All the features described in this chapter are not used. The design and organization of this Logical view can be used as a sample to create the one a user requires. Then, using the other functions demonstrated in this book (situations, advice, links, and so on), a Logical view that meets all the supervision requirements of a user is easy to build.

4.6 Logical view: Lessons learned

Following are some of the guidelines to follow when developing a Logical view:

- Keep the number of embedded levels small enough (three or four) because the user can get lost in the depth of a large tree
- Avoid too much information in a graphic view. Keep the graphic simple. Use the situation advice and their possibilities to allow the user get more information when required.

- Too much detailed information in workspaces and views require many changes if the IT infrastructure is modified. Generic supervision is enough if, at resolution time, the falling element is displayed.
- A Logical view is not a technical view. Users want to look at the relevant information to help them carry out their daily jobs.
- A Logical view that is always red is unsuitable. Users will stop trusting it. Adapt the alert definitions.
- The event console view is a useful tool because it shows all the properties of the events' activity. However, when using a Logical view, users want organized information and a quick way in which to resolve issues, which the event console is not adapted for.
- A graphical view can have a simple unicolored background. To create such a background, use any tool that generates a unicolored rectangle, and then save it as a jpg file or a png file in the user background directory of the TEP.
- Create a full screen graphic in the Logical view root. Create icons to represent the embedded levels. Do not set up any refresh interval when displaying this Logical view root. If an alert opens on any embedded level, this alert is propagated automatically up to the root graphic despite the fact that no automatic refresh is set up. This supervision method has the advantage of using few resources.
- Users can utilize the Navigator view to move from one level to one another level. A better approach is to hide the Navigator view and create a *graphic navigator approach*, that is, in each workspace, you create a graphic view (always at the same place in the workspace). Inside this workspace, insert your own icons that represent the different levels the users must go to. Then, define a link to give access to these levels.

Figure 4-73 shows a graphic navigator approach.

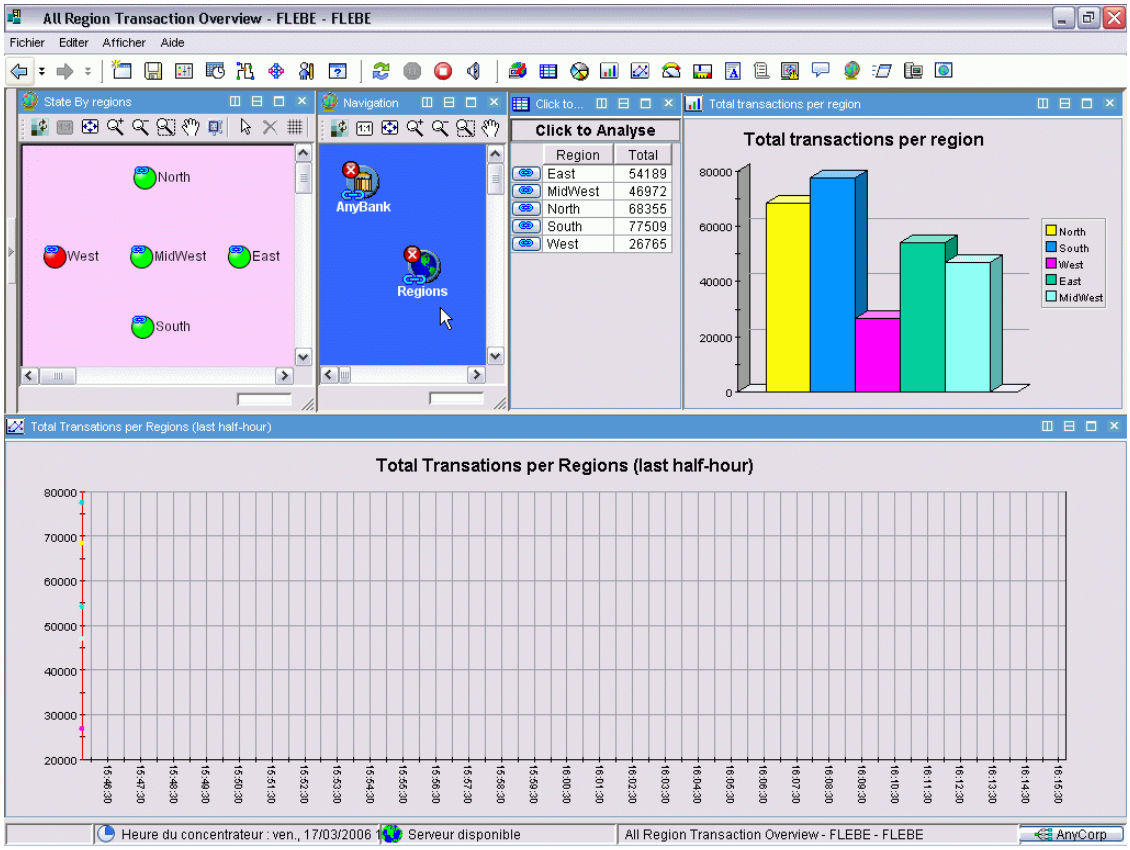


Figure 4-73 A graphic navigator approach

You can define a Logical view to perform the following functions, which are based on customers' realizations:

- Manage a CICSPlex

  To manage CICS regions over a sysplex, create a Logical view that shows the relationship between the terminal-owning region (TOR), application-owning region (AOR), and the others regions, and create a subtree for each collection of regions that manage a set of applications.

- Manage well-known problems

  Build a Logical view that includes all the information that indicates that one of your current known problems is starting, and add actions inside in order to quickly react and solve it.

- Handle the operation team's requirements

  Interview your operations team and ask them which data they require to have better control over the systems, and build Logical views to fulfill their requirements

- Reorganize the physical data

  Define a graphic that maps the physical emplacements of your systems, and then tag and drop each system into its corresponding place from the Physical view. This Logical view can be built quickly, and the existing alerts definition retained.

- Follow up on applications

  Build a Logical view by including all the systems that participate into a composite application. Set up alerts in order to follow the quality of service of these applications at each point.

- Determine the priorities of actual incidents

  Build a Logical view with levels such as Priority One, Priority Two, and so on. Associate a situation to its corresponding severity level. The operations team will know where to focus on first.

- Migrate from OMEGAVIEW

  Analyze your OMEGAVIEW organization. Reproduce each icon on an OMEGAVIEW display to a child in the Logical view tree, and then build situations to map the existing OMEGAMON thresholds and associate them at the right place. Then, add value using links, advice, and so on.

5

Scenarios and use cases

This chapter provides examples of Logical views and Physical views that the Smart Bank showcase uses to demonstrate its business proof points. The focus is on three use cases that help prove three different proof points and demonstrate the efficiency of IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6. Following is a list of these use cases:

- Core system transformation (service-oriented architecture)
- Optimization of IT resources (On/Off Capacity on Demand)
- Branch transformation and infrastructure simplification (heterogeneous view)

© Copyright IBM Corp. 2006. All rights reserved.

5.1 Core system transformation (service-oriented architecture)

This section shows the new service-oriented architecture model working at full volume. In the defined views, we will show how the Web service channels and the channels using the process mechanism meet service-level agreements at a higher enterprise level, by measuring the response time (the service class in WebSphere z/OS for the process running under WebSphere Business Integration Server Foundation). In this section, the focus is more on the actual components and mechanisms used in order to show, at a more detailed level, the process of monitoring and managing them. In this context, some of the techniques suggested by IBM to help enterprises transform the use of their core systems and leverage the key functionality that is otherwise difficult to expose, is also discussed.

5.1.1 Introducing the architecture through Tivoli Enterprise Portal

We created a number of Logical views using Tivoli Enterprise Portal (TEP) for this IBM Redbook project, which we used for the Smart Bank showcase. As a base, we started with the physical monitoring tree, listing the following key monitoring environments:

- IBM System z9 environment
- Linux environment
- AIX environment
- Windows environment

Within these high-level environments, the subsystem monitoring described in the earlier chapters exists.

In our case, we explored two key areas in a Smart Bank showcase environment:

- The Enterprise view of our bank, through which a more business-centric view of channels and the quality of service is explored. This is provided by the Workload Manager (WLM) running on IBM System z9 and Parallel Sysplex that provides a Logical view of the core systems running on IBM z/OS logical partitions (LPARs).
- The more operational Physical view in order to show how to monitor and manage the complex heterogeneous environment

We constructed the view shown in Figure 5-1 to provide links and alerts to these two trees of Logical views. Many of the figures in this chapter are live screen captures taken from the following:

- The TEP running with some operational workload over the banking channels, ATM, Internet, and Branch
- The process channel created for this demonstration to show the service-oriented architecture model, with additional ATM, Internet, and Branch workload

Green indicates that all the systems are running according to the service-level agreements set up in the TEP, or are simply up and running. Red indicates that there is a problem that requires attention. Figure 5-1 shows that there is a problem in the Operations view.

For core systems transformation, we decided to take the Enterprise route, and therefore, double-clicked the Enterprise alert to link to the next level of the logical tree.

Throughout this project, we used the TEP option of displaying the blue chain link symbol on the alert in order to differentiate between alerts with links attached and those without links.

Figure 5-1 Smart Bank showcase: The opening window

In order to provide a recognizable view of a typical retail bank, we chose to use a simplified Logical view of a retail bank, on which we superimposed the alerts for our key proof points and channels. Figure 5-2 shows the details of the TEP Enterprise View window. On the left, the channel alerts representing our multichannel environment are visible. When the workload is injected at different volumes during the course of the demonstration, these alerts provide a high-level view of whether our infrastructure is able to maintain the quality of service we have agreed upon with our business channels.

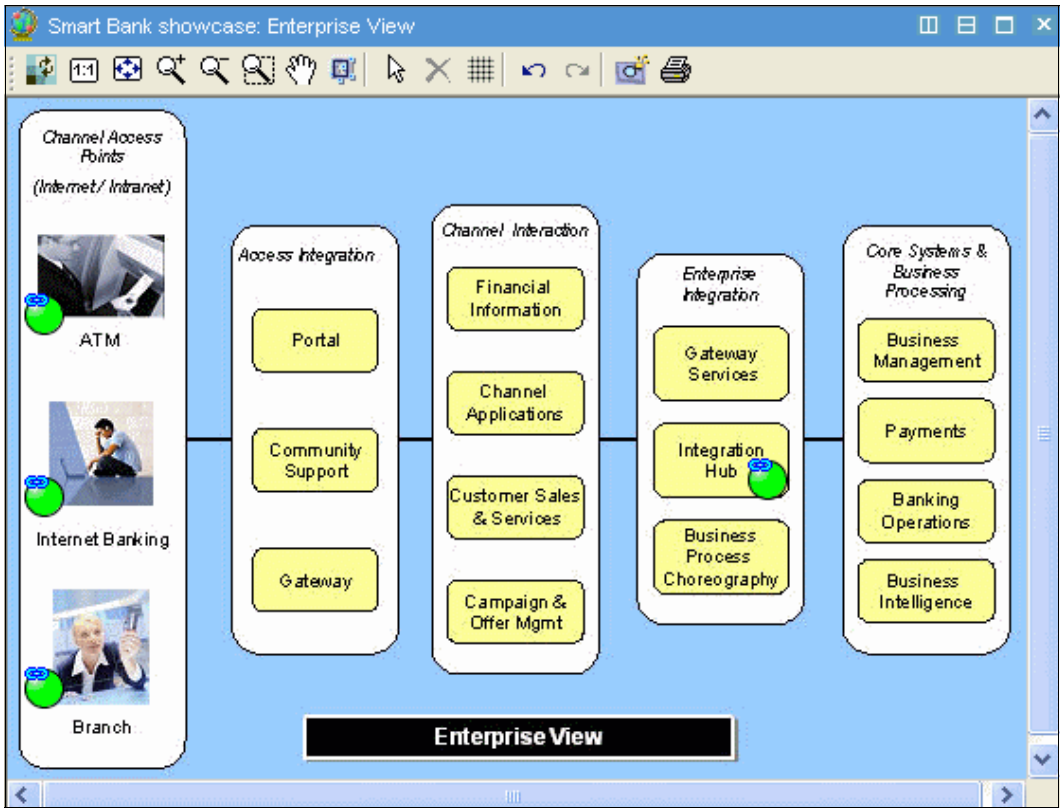


Figure 5-2 The Enterprise view

We added another alert over the Integration Hub box through which we linked to our service-oriented architecture view.

5.1.2 Service-oriented architecture view

The agents used in the Logical view (Figure 5-7 on page 266) come from the IBM Tivoli Composite Application Manager for Service-Oriented Architecture product that is installed on z/OS:

- Service management
- Service management environment

Details about how this workload is injected is provided at this point of the demonstration for a better understanding of the Logical views described here. During a demonstration, these explanations form a part of the presentation content.

Multichannel workload evolution

The workload shown in Figure 5-7 on page 266 is our multichannel workload being injected as individual Web services, using the SOAP/Hypertext Transfer (HTTP) protocol. This represents the simulation of our ATM and Internet channels, by injecting straight into the WebSphere Application Server on z/OS. Table 5-1 shows the way our core system evolved.

Note: Each step referenced in Table 5-1 is associated with one or more enterprise archive files (EAR files) deployed to the WebSphere Application Server for the different integration styles to coexist.

Table 5-1 Integration techniques

Step	Description	Technical details
1	Multichannel workload	**Client:** Struts-based Java 2 Platform, Enterprise Edition (J2EE) architecture, where we injected (using HTTP) and tested (using a Struts client through JavaServer Pages). Each operation calls an action bean that is specific to the type of application programming interface (API) on the core system. The action bean corresponds one-to-one with a stateless session bean. **Integration:** The session bean invokes the core system proprietary mechanism by invoking the method APICALL. We re-engineered the interface to use Java Message Service (JMS) and connection pools to put to the WebSphere MQ queue, instead of using the native Fidelity Java classes with the MQ API commands. We did this to adopt a more open approach and make the interface more efficient. Using GET WAITs, we created a synchronous implementation of this asynchronous mechanism. The core system uses its standard technique to launch the client Information Control System (CICS) transactions using Dynamic Transaction Routing across the CICSplex.
2	Web service multichannel workload	**Client:** The Struts framework action beans do not lend themselves well to the creation of Web services. The typical arguments for the Struts action beans are not particularly useful for a Web service call (HTTP request, HTTP response). Consequently, we created a new set of specific action beans that receive arguments corresponding to the input parameters for the core system API (delivery system, arrangement number, user ID, amount, and so on). We then created the Web service from these new action beans. **Integration:** The same as step 1.

Step	Description	Technical details
3	Service-oriented invocation of core system	**Client:** A completely new client. This client uses an action bean to call the Java bean proxy through a normal Java call that is formed during the creation of what is called an “Enterprise service”. The financial operations transactions that are called balance inquiry, transfers, and so on, remain the same as in step 1 and step 2. **Integration:** A completely new interface. Here, we re-engineered the interface technique to the core system, and instead of using the proprietary mechanism provided by the core system, we used the COMMAREA in CICS to interface to the same API. We created the “Enterprise service” definition directly from the COMMAREA, using the WebSphere development tool available with WebSphere Studio Application Developer Integration Edition V5 and used CICS Transaction Gateway V5 as the J2EE Connector mechanism (part of the J2EE Connector architecture [JCA]). Using this tool, a Wizard that drives the creation of the Web Services Description Language (WSDL) components, the JCA connector, and the Java bean proxy are created. At this point, we could have created a Web service from the Java bean proxy. However, we moved to step 4. **Note:** Another method would be to involve SOAP/HTTP directly to the core system.
4	Service-oriented invocation of a process	**Client:** Similar client as with step 3, in that, we can either call the Java bean proxies created from the binding to the process or the process EJB directly through Remote Method Invocation over InterOrb Protocol (RMI/IIOP). The bindings that are available are: ► SOAP/HTTP using the WSDL generated from the process ► SOAP/JMS using the WSDL generated from the process (not used in our project) ► RMI/IIOP using the WSDL generated from the process ► RMI/IIOP direct calling of the process EJB **Integration:** We incorporated the “Enterprise service” into a process and integrated this service with other services, which can be other Enterprise services or Web Services. The other services perform the near real-time update of the analytical database for different business purposes. These are Web services that have been generated from the Java bean that performs the JMS writes to the queue.

The figures that follow illustrate steps 1, 2, 3, and 4 described in Table 5-1.

Table 5-2 lists the acronyms used in these figures.

Table 5-2 List of acronyms used in the diagrams

Acronym	Description
AOR	Application-owning region in CICS
DTR	Dynamic transaction routing (CICSPlex)
TRAN	CICS Transaction
WMQ	WebSphere MQ
JMS	Java Message Service

MDB	Message-driven bean
JSP	JavaServer Pages
DPL	Dynamic Program link (CICSPlex)
PGM	Program (CICS)
CTG	CICS Transaction Gateway
JDBC	Java Database Connectivity
JCA	J2EE Connector architecture
EJB	Enterprise JavaBeans
TOR	Terminal-owning region in CICS

Figure 5-3 shows step 1 (described in Table 5-1), with the initial multichannel workload on WebSphere on z/OS.

Note: The same principle applies to the J2EE applications deployed on WebSphere in Linux in the Blades and in the IBM z/VM guests that represent the branch servers.

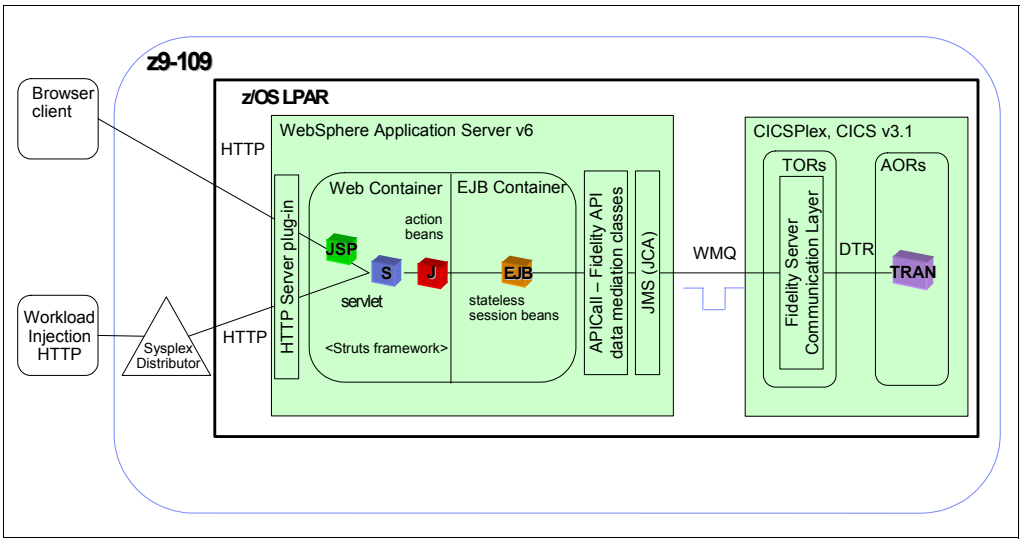


Figure 5-3 Initial multichannel workload (step1)

Figure 5-4 shows step 2 (described in Table 5-1), which we used in this book to show the actual Web services running through the infrastructure.

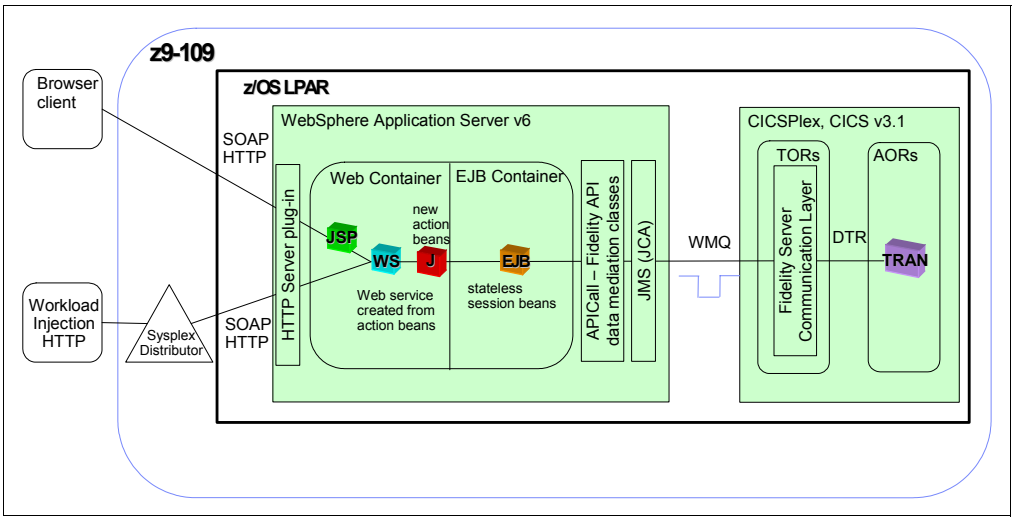


Figure 5-4 Multichannel workload injected with Web services (step 2)

Figure 5-5 shows step 3 (described in Table 5-1), our interim step to use the CICS Transaction Gateway and the strategic software tooling provided by WebSphere and IBM Rational®.

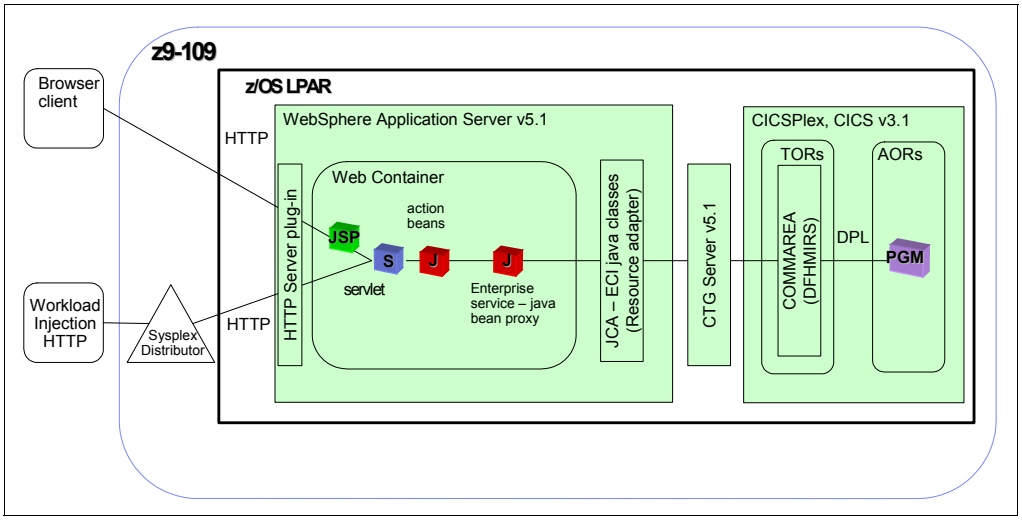


Figure 5-5 Service-oriented invocation of the core system (step 3)

Note: There are many integration mechanisms that can be used. We chose the CICS Transaction Gateway mechanism. Another equally viable and strategic choice is SOAP/HTTP direct to CICS.

Figure 5-6 shows step 4 (described in Table 5-1), which is our full process integration step that uses the Enterprise services created in step 3, but combines them with other services and uses the Process Choreographer provided by the WebSphere Business Integration Server Foundation. The run-time element of the process is an EJB. It is this workload that we monitor, as described in 5.1.4, “Process integration” on page 269.

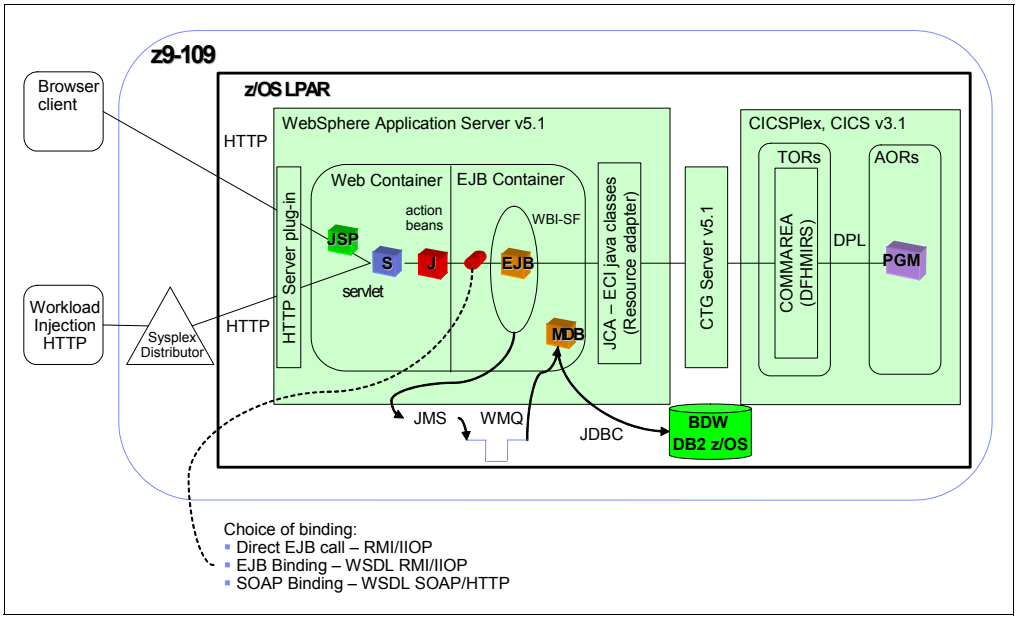


Figure 5-6 Service-oriented invocation of a process (step 4)

Our service-oriented Logical views

Figure 5-7 shows the results of running the workload described in step 2. IBM Tivoli Composite Application Manager for SOA is able to provide a Logical view, identifying the individual Web services through the use of the Java API for Extensible Markup Language-based Remote Procedure Calls (JAX-RPC).

On the right-hand side, the response time in milliseconds for each individual Web service is displayed. This response time includes transaction routing from WebSphere z/OS to CICS through WebSphere MQ, accessing and updating DB2 on z/OS through the Fidelity Corebank application, and back again.

Figure 5-7 is a JPEG image loaded as a graphic into one of the TEP frames, with alerts superimposed as with the initial Smart Bank and Enterprise views. A diagram that conforms to the IBM SOA reference architecture is displayed here. As indicated earlier, we use the term *Integration Hub* because we do not have the WebSphere V6 implementation of the Enterprise Services Bus implemented in our project environment.

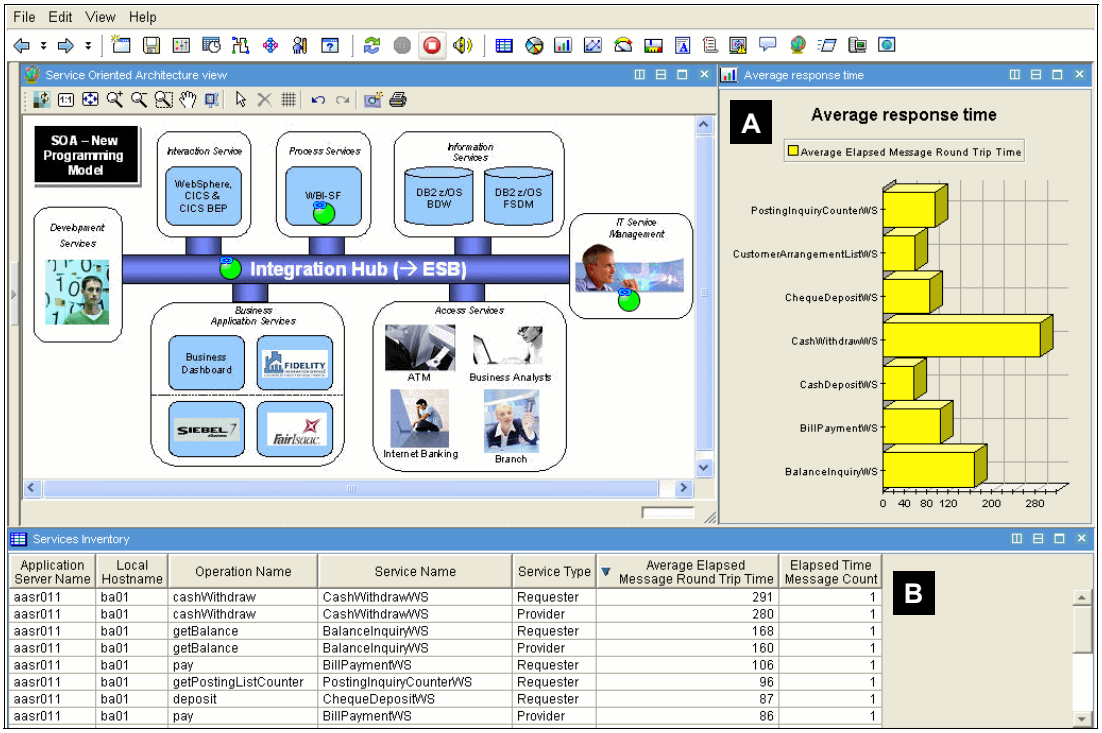

File Edit View Help
Service Oriented Architecture view
SOA – New Programming Model
Interaction Service
WebSphere, CICS & CICS BEP
Process Services
WBI-SF
Information Services
DB2 z/OS BDW
DB2 z/OS FSDM
Development Services
IT Service Management
Integration Hub (→ ESB)
Business Application Services
Business Dashboard
FIDELITY
SIEBEL 7
FairIsaac
Access Services
ATM
Business Analysts
Internet Banking
Branch
Average response time
A
Average response time
Average Elapsed Message Round Trip Time
PostingInquiryCounterWS
CustomerArrangementListWS
ChequeDepositWS
CashWithdrawWS
CashDepositWS
BillPaymentWS
BalanceInquiryWS
0 40 80 120 200 280
Services Inventory
B
Application Server Name | Local Hostname | Operation Name | Service Name | Service Type | Average Elapsed Message Round Trip Time | Elapsed Time Message Count
aasr011 | ba01 | cashWithdraw | CashWithdrawWS | Requester | 291 | 1
aasr011 | ba01 | cashWithdraw | CashWithdrawWS | Provider | 280 | 1
aasr011 | ba01 | getBalance | BalanceInquiryWS | Requester | 168 | 1
aasr011 | ba01 | getBalance | BalanceInquiryWS | Provider | 160 | 1
aasr011 | ba01 | pay | BillPaymentWS | Requester | 106 | 1
aasr011 | ba01 | getPostingListCounter | PostingInquiryCounterWS | Requester | 96 | 1
aasr011 | ba01 | deposit | ChequeDepositWS | Requester | 87 | 1
aasr011 | ba01 | pay | BillPaymentWS | Provider | 86 | 1

Figure 5-7 SOA model view: IBM Tivoli Composite Application Manager V6 for SOA

Table 5-3 shows the base queries that we used to create the Logical view from the assigned agents in Figure 5-7.

Table 5-3 SOA model queries from the assigned agents

Workspace	Assigned agent: Query	Assigned managed system
A	Services Management Agent Environment /Services Inventory/Query: *Performance Summary*	WebSphere on z/OS
B	Services Management Agent Environment /Services Inventory/Query: *Services Inventory*	WebSphere on z/OS

At this stage of the showcase, we demonstrated that we have successfully exposed the selected core system functions as services using open standards. Now it is up to the institution whether these functions are invoked as Web Services as shown here in order to demonstrate IBM Tivoli Composite Application Manager for SOA or as enterprise services using the WSDL definitions to help integrate the core system functionality.

After using IBM Tivoli Composite Application Manager for SOA that is deployed here on z/OS, for a better understanding of our services, we select the IT Services Management link in the window shown in Figure 5-7. This displays the window displayed in Figure 5-8. Here, the size of the individual service messages and the number of messages flowing through the system are monitored.

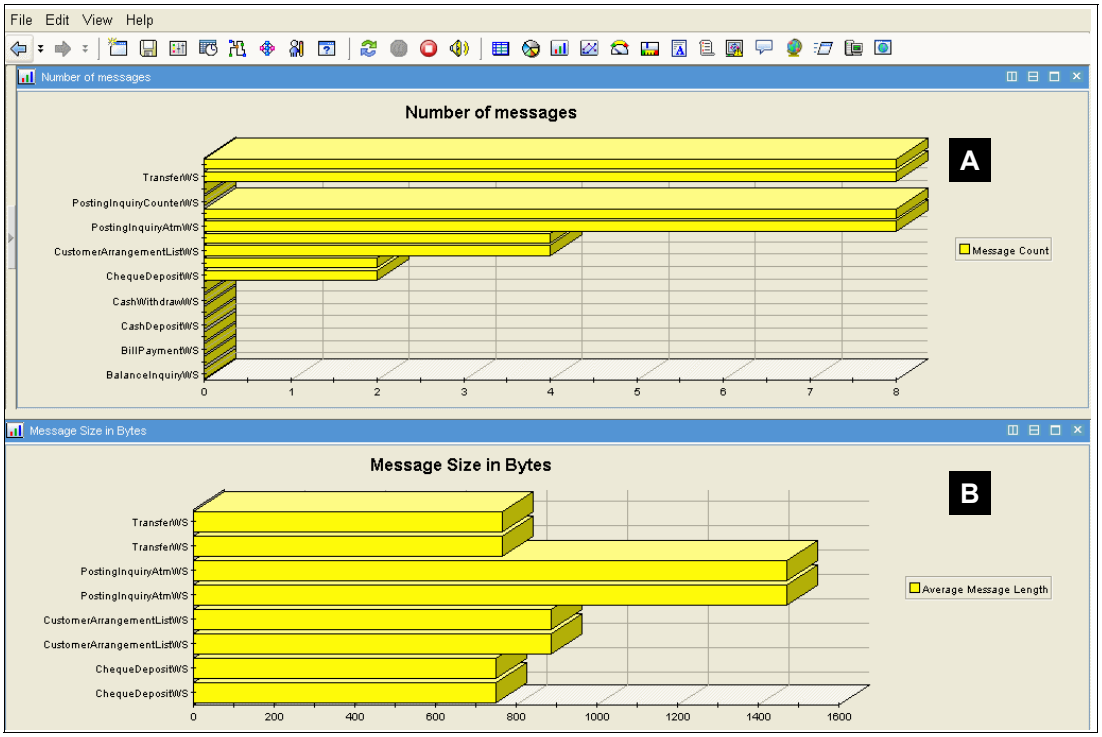


Figure 5-8 Service management: IBM Tivoli Composite Application Manager V6 for SOA

Table 5-4 shows the base queries that we used to create the Logical view from the assigned agents in Figure 5-8.

Table 5-4 Service management queries

Workspace	Assigned agent: Query	Assigned managed system
A	Services Management Agent Environment /Services Inventory/Query: *Message Summary*	WebSphere on z/OS
B	Services Management Agent Environment /Services Inventory/Query: *Message Average*	WebSphere on z/OS

5.1.3 Integration hub: WebSphere MQ

From the alert in the integration hub in the SOA model view (Figure 5-7), we linked to the WebSphere MQ Logical view displayed in Figure 5-9.

WebSphere MQ as a product provides a crucial integration mechanism for our project and is also a foundation for the Enterprise Service Bus. We left IBM Tivoli Composite Application Manager for SOA monitoring and used the IBM Tivoli Monitoring V6.1 agents to construct our Logical views.

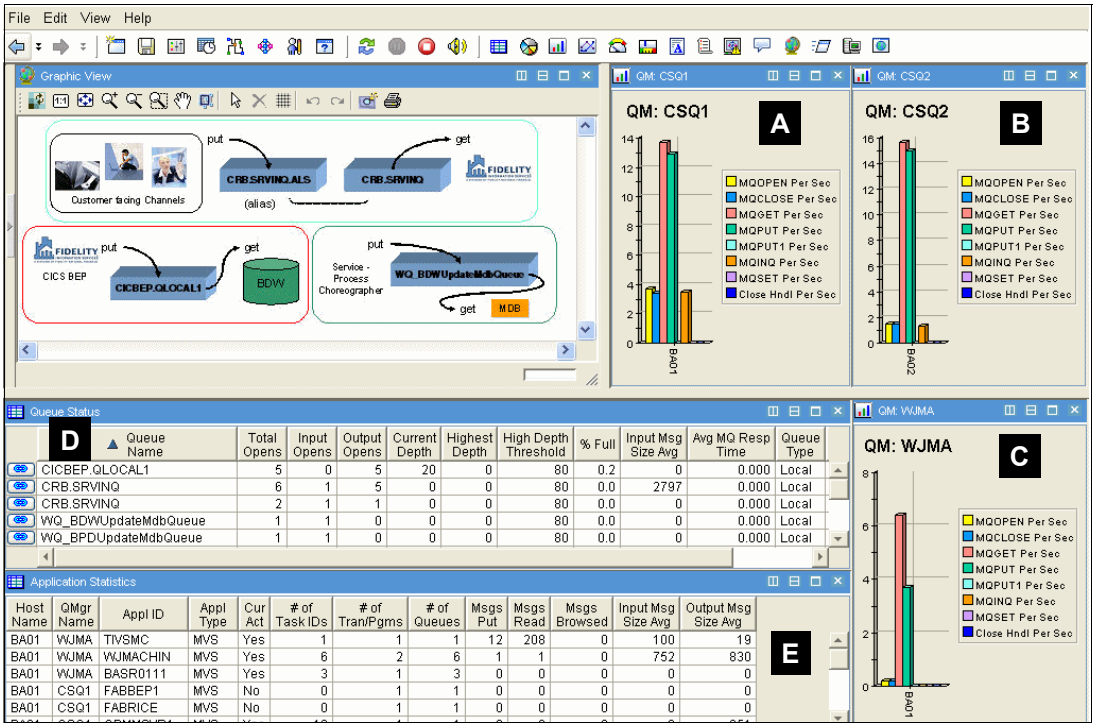

File Edit View Help
Graphic View
Customer facing Channels
CRB.SRVINQ.ALS
(alias)
CRB.SRVINQ
CICS BEP
CICBEP.QLOCAL1
BDW
Service - Process Choreographer
WQ_BDWUpdateMdbQueue
MDB
QM: CSQ1
A
QM: CSQ2
B
MQOPEN Per Sec
MQCLOSE Per Sec
MQGET Per Sec
MQPUT Per Sec
MQPUT1 Per Sec
MQINQ Per Sec
MQSET Per Sec
Close Hndl Per Sec
BA01
BA02
Queue Status
D
Queue Name | Total Opens | Input Opens | Output Opens | Current Depth | Highest Depth | High Depth Threshold | % Full | Input Msg Size Avg | Avg MQ Resp Time | Queue Type
CICBEP.QLOCAL1 5 0 5 20 0 80 0.2 0 0.000 Local
CRB.SRVINQ 6 1 5 0 0 80 0.0 2797 0.000 Local
CRB.SRVINQ 2 1 1 0 0 80 0.0 0 0.000 Local
WQ_BDWUpdateMdbQueue 1 1 0 0 0 80 0.0 0 0.000 Local
WQ_BPDUpdateMdbQueue 1 1 0 0 0 80 0.0 0 0.000 Local
Application Statistics
Host Name | QMgr Name | Appl ID | Appl Type | Cur Act | # of Task IDs | # of Tran/Pgms | # of Queues | Msgs Put | Msgs Read | Msgs Browsed | Input Msg Size Avg | Output Msg Size Avg
BA01 WJMA TIVSMC MVS Yes 1 1 1 12 208 0 100 19
BA01 WJMA WJMACHIN MVS Yes 6 2 6 1 1 0 752 830
BA01 WJMA BASR0111 MVS Yes 3 1 3 0 0 0 0 0
BA01 CSQ1 FABBEP1 MVS No 0 1 1 0 0 0 0 0
BA01 CSQ1 FABRICE MVS No 0 1 1 0 0 0 0 0
E
QM: WJMA
C

Figure 5-9 WebSphere MQ view: Part of the integration hub

Table 5-5 shows the base queries that we used to create the Logical view from the assigned agents in Figure 5-9.

Table 5-5 WebSphere MQ queries

Workspace	Assigned agent: Query	Assigned managed system
A,B,C	MQSERIES/Managers/Query: *Message Manager Performance*	Queue Managers CSQ1, CSQ2, and WJMA respectively
D	MQSERIES/Queue Statistics/Query: *Queue Statistics*	Queue Managers running on BA01 and BA02 z/OS LPARs
E	MQSERIES/Application Statistics/Query: *Application Statistics*	Queue Managers running on BA01 and BA02 z/OS LPARs

For demonstration purposes, we included other JPEG graphical views (not illustrated here) that illustrate the key queues that our system uses:

- CRB.SRVINQ and alias for the multichannel integration (step 1 and step 2 in Table 5-1 on page 261) (CSQ1 Queue Manager and CSQ2 Queue Manager)
- CICBEP.QLOCAL1 for the near real-time update of the analytical database through the CICS BEP product (CSQ1 Queue Manager)
- WQ_BDWUpdateMdbQueue and WQ_BPDUpdateMdbQueue for the near real-time update of the analytical database through the process mechanism (step 4 in Table 5-1 on page 261) (WJMA Queue Manager)

On the right-hand side of the Logical view, a Queue Manager view of the WebSphere MQ activity, CSQ1, CSQ2, and WJMA corresponding to the frames A, B, and C is displayed.

In Table 5-5 (frame D), the queue depth on our queues is displayed. This instantly alerts if the queue depth begins to increase beyond what is expected, indicating that the application or mechanism that is listening to the queue in order to process the message, may be experiencing a problem.

In frame E, the applications are displayed. These are the tasks started on z/OS that are attached to the different queue managers processing against the different queues.

The applications running against the WJMA queue manager are seen as being active. These are our WebSphere z/OS servant region and JMS Server running our Process Choreographer. The queue depth for the WQ_* queues is zero, indicating that with the current rate of throughput on the WJMA queue manager, there are no problems relating to WMQ with the Process Choreographer.

However, there is a queue depth of 20 on queue CICBEP.QLOCAL1, which has not triggered the threshold, but is indicative that there are 20 messages waiting to be updated to our analytical database. The application task FABBEP1 is stopped. It is this task that must be a long-running task in order to poll the CICBEP.QLOCAL1 queue and update our banking data warehouse.

From this Logical view, a lot of information about the queue mechanisms that are central to our integration architecture and process integration is derived.

5.1.4 Process integration

At this stage, we tried to get a more detailed view of the process that is running at volume (step 4 in Table 5-1). We run this on WebSphere V5.1 with the Process Choreographer (WBI SF). In WebSphere V6, this is renamed as the Process Server, which, at the time of writing this book, was *not* available on z/OS.

From the SOA model Logical view (Figure 5-7), we can link to the Process Integration view shown in Figure 5-10.

After the IBM Tivoli Monitoring products are installed, the capability to gain a Business Process Execution Language (BPEL) view of the process is not present. However, with an understanding of the underlying mechanisms and a graphical view of the process taken from WebSphere Studio Application Developer - Integration Edition V5 (WSAD-IE), we construct a Logical view.

Important: The same technique that is used to integrate to CICS in our Smart Bank showcase demonstration can be used with IMS. Instead of using the CICS Transaction Gateway (CTG) as the J2EE Connector, we used IMS Connect. Both CICS and IMS support the use of direct Web service integration using the SOAP/HTTP protocol.

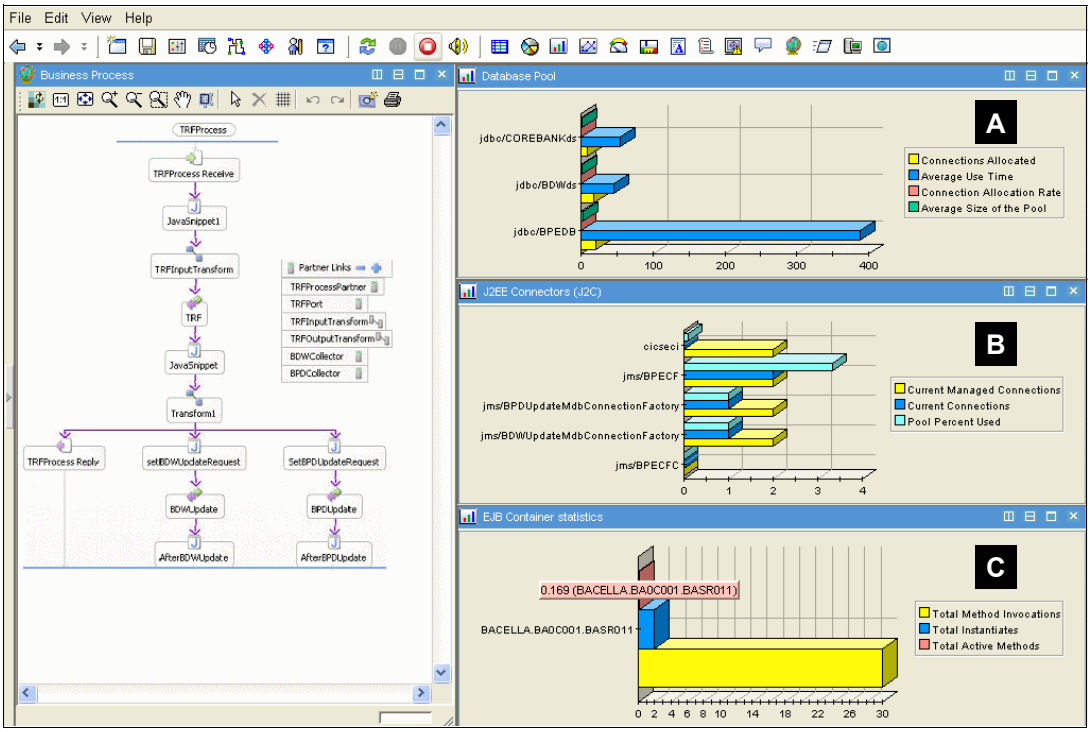


Figure 5-10 Process component view

Table 5-6 shows the base queries that we used to create the Logical view from the assigned agents in Figure 5-10.

Table 5-6 Process view queries

Workspace	Assigned agent: Query	Assigned managed system
A	J2EE Server /DB Connection Pools for OS/390/Query: *DB Connection Pools*	WebSphere on z/OS
B	J2EE Server /J2EE Connector connection pools for OS/390/ Query: *J2EE Connector connection pools*	WebSphere on z/OS
C	J2EE Server /EJB Containers for OS/390/Query: *EJB Containers*	WebSphere on z/OS

Figure 5-10 provides a graphical JPEG view of the process. The Transaction Reporting Activity (TRF) issues the service call to CICS, invoking our core system functionality. On receiving a response from our core system, the process performs the following actions within the same unit of work:

- Web Service call (update of the analytical database [banking database warehouse])

  The process extracts the relevant information from the core system response and writes it as a message to a queue using JMS. An MDB is triggered by the presence of a message in the queue and it performs the insert through JDBC to the banking database warehouse (DB2 on z/OS).

- Web Service call (update of banking database warehouse for business performance)

  The process extracts the relevant information from the core system response and enriches this data with more information retrieved by a Structured Query Language (SQL) call to the financial services data model (FSDM) database (DB2 on z/OS). This information is then written as a message to a different queue using JMS. An MDB that is triggered by the message updates our BDW. This is a different business reason from our other analytical database update, and affects different tables.

- Reply to the process or service requestor after the entire process is completed successfully

This process can be bound in a number of different ways, as already discussed in Table 5-1. For this book, we used the simple RMI/IIOP call of the process EJB.

Our Logical view provides the following operational information about this process:

- With the JDBC connections information, calls to the operational database (FSDM) and our analytical database for the BDW update and for the business performance dashboard update are seen.
- With the J2EE Connector (J2C) view, the CICSSECI connection pool, which refers to the CICS Transaction Gateway and our service to invoke our core systems functionality, are seen. As we write to the queues for asynchronous update of our BDW database, the JMS activity is also seen.
- Finally, in the third view, the EJB activity in WebSphere on z/OS, which represents the method invocations of our process, is seen.

5.1.5 Summary

After the Logical views are constructed, our service-oriented architecture and our integration mechanisms are seen. The use of open standards is the key to allowing such an integration. The use of TEP, IBM Tivoli Monitoring V6.1, and IBM Tivoli Composite Application Manager for SOA help in monitoring and managing the service-oriented environment.

The choice of WebSphere on z/OS as the platform for our Integration Hub and Process Choreographer was taken based on a number of factors, some of which are listed here:

- Our core systems functionality was already on IBM System z in CICS. In our case, this meant that we could benefit from the efficiencies and richer integration functionality of being on the same platform.
- The underlying quality of service or the IBM System z9 platform. Many see SOA as the strategic architectural way forward. Many are also of the opinion that in order to truly benefit from this, an Enterprise view and scope is required. Consequently, the integration and provision of these services becomes a mission-critical imperative that the IBM System z9 platform is clearly well positioned to provide.

- The use of the new IBM System z Application Assist Processor (zAAP) Java workload-specific processors on IBM System z9 makes the cost of running a Java workload on the platform much more attractive.

In the case of our retail bank, this allows them to address the key aspects of revenue growth and integration identified at the beginning, and helps them rapidly create new products, reducing the time to market while leveraging what they already have.

After isolating the selected key core system functionality as services, we selected the technical binding to suit the type of integration required. By using processes and the Process Choreographer (WBI SF with WebSphere V5 and Process Server with WebSphere V6), new business functions can be created quickly and flexibly, combining core system functionality with other functionality that can be exposed with WSDL definitions. Multiple core systems and potentially external services can be integrated together to create what are called composite applications. This can also be referred to as services. The Process Choreographer or Process Server handle hierarchies of processes and services.

5.2 Optimization of IT resources (On/Off Capacity on Demand)

In this proof point, we show how our operational environment is being stretched from a resource utilization point of view. We inject a workload that is unusually high and corresponds to not only the normal peak load at lunch time, but also has additional workload due to the success of our new product launch. Our bank experiences an unusual increase in workload. If our system cannot respond quickly to maintain the quality of service to the client-facing channels, business will be lost. The alerts on our Enterprise view turn yellow or red when our channels start experiencing bad response times.

In our demonstration, we perform an On/Off Capacity on Demand to our System z9-109 environment, which provisions more processing capacity to our environment, that is, it automatically makes available more processors such as central processors (CPs), zAAPs, and Integrated Facility for Linux (IFLs) for our workload to use.

An important point here is that without virtualization, our workload will not be able to take advantage of this extra processing resource. The fact that our mission-critical applications run in LPARs on z/OS and in z/VM, which share physical CPs, makes this possible. Our additional CPs are recognized by the environment, and within 3 - 5 minutes, we have our system running with the quality of service we agreed on.

The Hardware Management Console (HMC) for the IBM System z9-109 is used to initiate the On/Off Capacity on Demand, entering the pre-prepared policy number, which is verified over the Internet by the IBM Resource Link™ in Poughkeepsie, NY, USA. The Resource Link identifies the policy and enables the physical processors on our IBM System z9-109. Our bank is billed for this on a daily basis until such time as they are deprovisioned. Through the HMC, the number of CPs that have been activated and the monitoring of the On/Off Capacity on Demand process are shown.

To have a more transparent billing of CPU utilization within an enterprise on the IBM System z9-109 is something that a number of clients are looking to implement. Under this, they pay for what they use, and they, along with IBM, enter into an overall system-wide agreement that covers hardware, software, and services. This type of agreement is customized and comes under an Open Infrastructure Offering (OIO) agreement.

In this proof point, we take a different path down the Logical view tree, and move down the Operations track. We link to the Operations view from our opening Smart Bank screen (Figure 5-1 on page 259).

5.2.1 The Operations view

Figure 5-11 shows the Operations view. Red alerts in the TEP are cascaded down the tree hierarchy. In this view, the fact that the red alert is related to systems management is seen.

In this Logical view, we created a JPEG graphical image depicting our physical environment. We added a number of alerts that reflect the status of the underlying subsystems and platforms. Not all of them have links, but they could. We focused only on those aspects that concern our project. On the top right-hand side are a series of navigational alerts and links that help us navigate between our different Logical views during a demonstration.

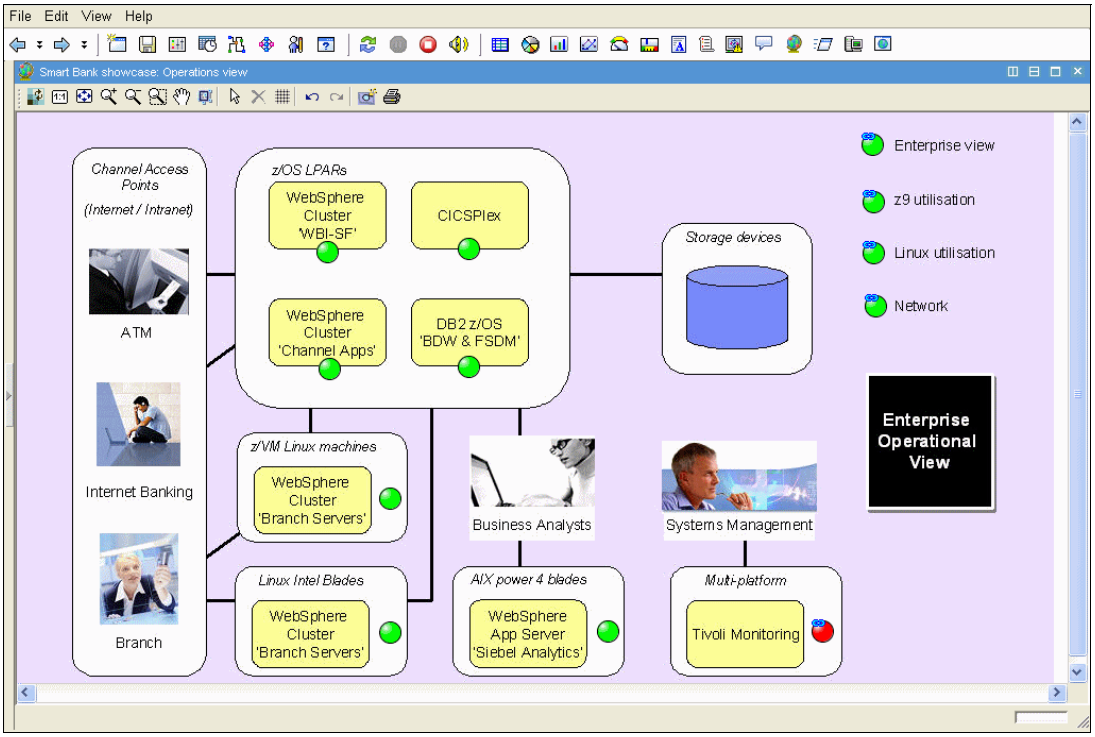


Figure 5-11 The Operational view

Table 5-7 shows the base queries that we used to create the Logical view from the assigned agents in Figure 5-11.

Table 5-7 Operational view queries

Alert	Assigned agent	Assigned managed system
WebSphere Cluster - WBI SF	z/OS Systems/WebSphere Application Server/J2EE Server /Application Server Instance	WebSphere for z/OS V5 cluster with WBI SF on BA01 and BA02 z/OS LPARs
WebSphere Cluster - Channel Applications	z/OS Systems/WebSphere Application Server/J2EE Server /Application Server Instance	WebSphere on z/OS V6 on BA01 and BA02 z/OS LPARs
CICSPlex	z/OS Systems/CICS/Region Overview	AOR and TOR regions on BA01 and BA02 z/OS LPARs
DB2 z/OS (BDW and FSDM)	z/OS Systems/DB2/System Status	DW01 and CB01 DB2 systems on BA01 and BA02 z/OS LPARs

Alert	Assigned agent	Assigned managed system
BladeCenter Branch Servers	► Linux Systems/Linux OS ► Linux Systems/WebSphere Application Server Agent ► Linux Systems/Universal Agent	► Blade 2: Linux OS and Universal Agent ► Blade 3: Linux OS ► Blade 4 and Blade 5: Linux OS and WebSphere Application Server
z/VM Branch Servers	► Linux Systems/Linux OS ► Linux Systems/WebSphere Application Server Agent ► Linux Systems/Universal Agent	► lnx1: Linux OS and Universal Agent ► lnx7: Linux OS ► lnx4 and lnx5: Linux OS and WebSphere Application Server
Siebel Analytics WebSphere	► UNIX Systems/UNIX OS ► UNIX Systems/WebSphere Application Server Agent	SAFSS POWER4 blade with AIX
Tivoli Monitoring	Link	Workspace
Network	Link	Workspace
z9 utilization	Link	Workspace
Linux utilization	Link	Workspace
Enterprise view	Link	Workspace

Figure 5-12 shows the main Edit Navigator view in the TEP that was used to assign the different physical agents to the alerts shown in Figure 5-11. The different physical and logical machines defined to our system and subdivided by the operating system, are seen.

Note: The Linux guest machine on z/VM, lnt1, was not used in the context of this book.

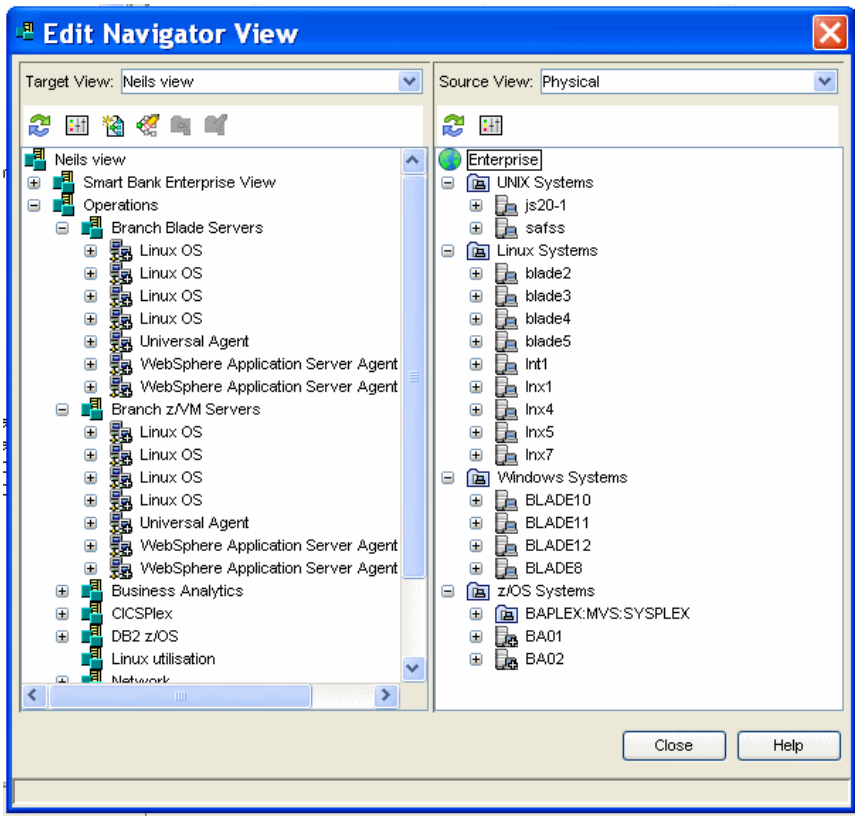


Figure 5-12 TEP Edit Navigator view

We now have a high-level view of the infrastructure and know that a potential problem exists. We used the link provided in the systems management alert to dig deeper into the problem. The same actions are performed throughout the Physical view tree, but the business and operations context are not so obvious.

5.2.2 Monitoring the monitoring environment

Figure 5-13 shows a Logical view of the monitoring environment. This is a simplified view of the monitoring infrastructure, with alerts superimposed on a JPEG image. None of these alerts are links, but they show the status of each of the components. Figure 5-13 shows that it is the Windows environment of the TEP that is causing the alert.

In the TEP, we created situations to detect whether the Moveable Hub is active on one of the z/OS partitions or not. Instead of sending a red or yellow alert if it is not detected in the other LPAR, an informational event is instead sent, which shows up as a blue diamond.

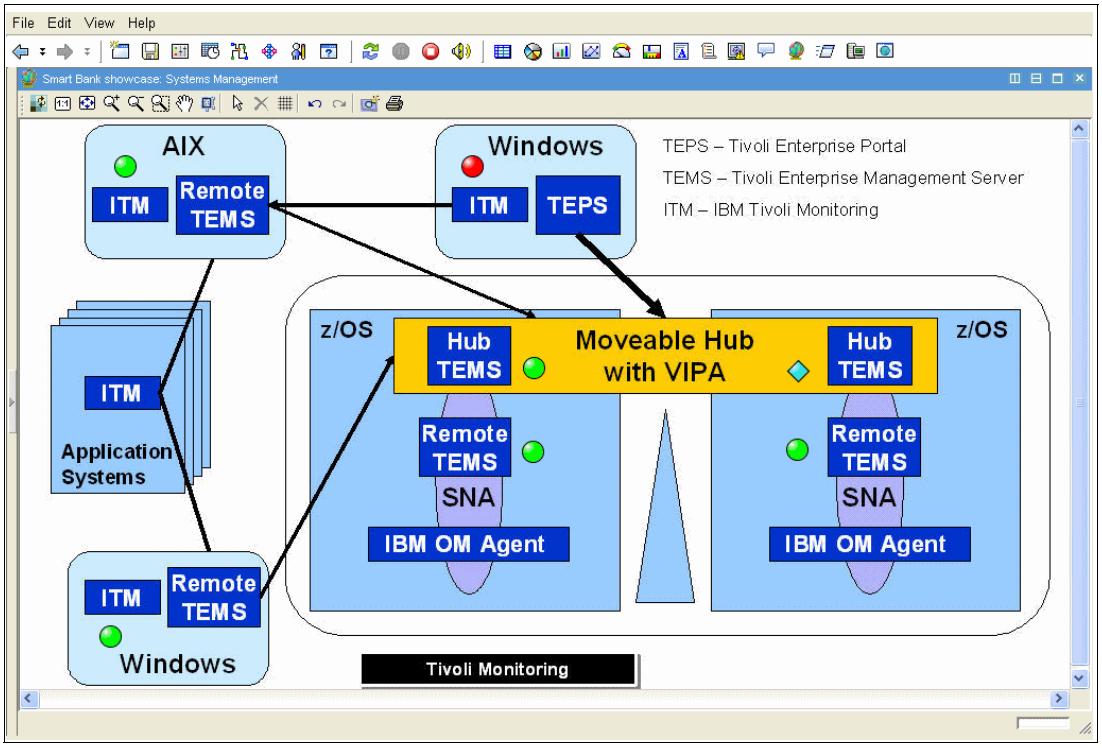


Figure 5-13 Monitoring the monitoring environment

Table 5-8 show the base queries that we used to create the Logical view from the assigned agents in Figure 5-13.

Table 5-8 Monitoring view queries

Alert	Assigned agent and situations	Assigned managed system
AIX Tivoli Enterprise Monitoring Server Remote	UNIX Systems/UNIX OS	js20-1 POWER4 blade
Windows Tivoli Enterprise Monitoring Server Remote	Windows Systems/Windows OS	Blade11
Windows Tivoli Enterprise Portal Server	Windows Systems/Windows OS Windows Systems/Universal Agent	Blade8
z/OS Moveable Hub	MVS System/MVS Address Space Situation: Jobname TIVSHUBT running on BA01 or BA02	z/OS LPARs BA01 and BA02. If job name is not found, informational state alert is triggered (blue).

Alert	Assigned agent and situations	Assigned managed system
z/OS Tivoli Enterprise Monitoring Server Remote	MVS System/MVS Address Space Situation: Jobname TIVSREMT running on BA01 & BA02	z/OS LPARs BA01 and BA02. If job name is not found, critical state alert is triggered (red).

To return to the Operations view, the Back icon on the toolbar at the top is selected.

5.2.3 Monitoring the network

From the Operations view, the network with the domain names displayed and the actual IP addresses hidden, can be seen. Figure 5-14 shows this view.

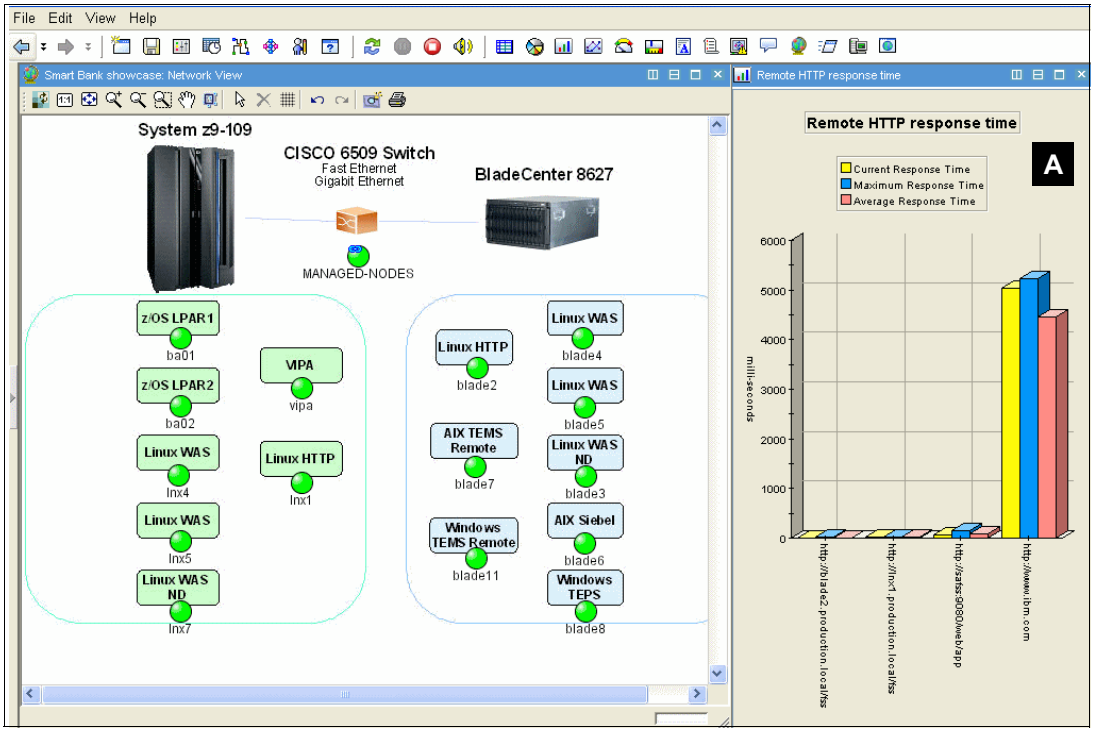


Figure 5-14 Network view of the operating environment

Table 5-9 shows the base queries that we used to create the Logical view from the assigned agents in Figure 5-14.

Table 5-9 Network view queries

Alert/ Workspace	Assigned agent, situations, queries	Assigned managed system
z9 and Blade nodes alerts	Situation created with: Universal Data Provider checking node name and node status. If offline, then critical state event is sent	Individual node specified for each situation

Alert/ Workspace	Assigned agent, situations, queries	Assigned managed system
A	Universal Data Provider/Internet /Managed URL/Query: *Managed URL*	► http://www.ibm.com (IBM US Web page) ► http://safss:9080/web/app (Siebel Analytics login) ► http://lnx1.production.local/fss (branch application) ► http://blade2.production.local/fss (branch application)
Managed Nodes	Link to Physical view Universal Data Provider/SNMP Manager /Managed Nodes	Blade8

In the left-hand side of Figure 5-14, the IBM System z9-109 environment is displayed, with the virtual IP address node denoting the sysplex distributor, the two z/OS LPARs, and the Linux guest machines on z/VM. The names below the alerts indicate the domain names. The MANAGED NODES alert is also a link to the Physical view. In the centre is the Blade Center environment, with the monitoring nodes, Linux branch nodes, and Siebel Business Analytics node.

To the right is the remote monitoring of the response times reported to the different nodes in the network. They are, from left to right:

- Linux HTTP server on an Intel blade
- Linux HTTP server on z/VM
- AIX Web server on a Power4 blade
- The IBM Web site, which shows the long response time because access was denied by the firewall

To return to the Operations view, the Back icon in the toolbar on top of the page is selected.

5.2.4 IBM System z9-109 utilization

Obtaining a Logical view of the IBM System z9-109 utilization is the main challenge for this proof point. The views discussed earlier are good for further discussions on the environment and help build a picture of the infrastructure deployed, and the capability of IBM Tivoli Monitoring V6.

The project development region used for this book is powered by an IBM eServer zSeries z990, whereas our target production platform is IBM System z9-109. Consequently, all the Logical view diagrams reference IBM System z9-109, whereas the actual machine being monitored is the z990, which did not have any zAAP or IFL processors defined.

We show more power and a greater variety of processors in our production environment. However, the principles and goals remain the same. Figure 5-15 provides an initial view of how this will look when we try and monitor all the subsystems, LPARs, and processors running on an IBM System z9-109. In workspace A, we see the CPU utilization for the application owning z/OS LPARs. The blue bar shows the zAAP utilization with its Java workloads in our production environment. This is used to provide details about the actual CPU utilization during a demonstration when more Java-intensive workloads are run. Workspaces B, C, D, and E show the breakdown of that CPU by some of the key subsystems running the application load.

Workspace B shows the Terminal-Owning region and Application-Owning region for our CICSplex. We instrumented the queries so that only those subsystems with activity are shown. Workspace C shows the WebSphere MQ activity, showing the Queue Manager task.

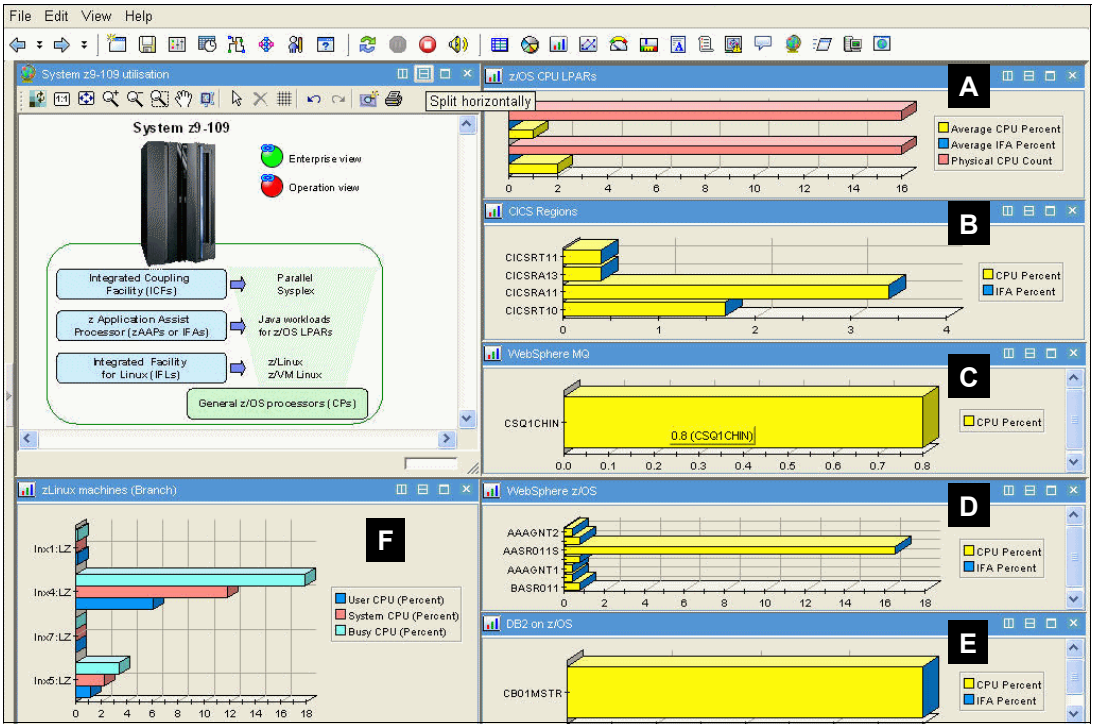


Figure 5-15 z9 utilization view

Table 5-10 shows the base queries used to create the Logical view from the assigned agents in Figure 5-15.

Table 5-10 z9 utilization queries

Workspace	Assigned agent query	Assigned managed system
A	MVS System/System CPU Utilization /Query: *System CPU Utilization*	z/OS LPARs BA01 and BA02
B,C,D,E	MVS System/Address space CPU utilization /Query: *Address space CPU utilization*	Generic job names that cover the started tasks from the various z/OS subsystems
F	Linux OS/Linux CPU/Query: *CPU*	Linux guests on z/VM

Workspace D then shows the WebSphere subsystems and workspace E the DB2 systems running. Finally, workspace F shows the WebSphere cluster running on z/VM, completing the picture of our application loads running on IBM System z9-109.

5.2.5 Summary

We show the basic CPU activity on our main Enterprise view so that, even as we monitor the service-level agreements of our client-facing channels, we can view the CPU utilization as we vary the volumes. Thus, this Logical view is a more technical view, showing the details.

When we perform the On/Off Capacity on Demand, the following points become obvious to the client:

- System resources can be dynamically added to the System z9-109 environment when required.
- The in-flight workload immediately takes advantage of this extra resource and maintains service-level agreements with business channels, and gains business.

Until now, a description of how IBM Tivoli Monitoring V6.1 provides enterprise-wide views of the infrastructure, while focusing on specific platforms, in our case, IBM System z9-109, was provided.

5.3 Branch transformation and infrastructure simplification (heterogeneous view)

In this proof point, we demonstrate to our clients the ability to simplify the distributed environment by consolidating our branch servers to Linux guests on z/VM. This is in keeping with a real-life request proposed by a client. The main business reason provided by the bank was reducing operational costs through centralization and simplification.

With IBM Tivoli Monitoring V6.1, we monitored this environment on z/VM with the branch workload we injected directly to the Linux branch servers.

This section shows the monitoring of a heterogeneous environment. Applications that adhere to the J2EE programming model, which are consequently platform-independent, can be deployed on a number of different operating systems. With Linux as an open operating environment, commonality of operating system across different platforms can be achieved. With open standards being adopted increasingly, the deployment choices for applications are based more on the quality of service required for business function, rather than hardware and software limitations.

We then arrived at a balance between the cost and the quality of service. We therefore, chose z/VM as our client example, as did our client's model, based largely on the requirement of reducing cost and increasing simplification. The z/VM environment, however, delivers many other key services.

The other deployment platform for our branch server application is the BladeCenter, which was also chosen for the infrastructure simplification qualities that it provides.

The IBM BladeCenter

An IBM BladeCenter is a high-density stack of servers that enables you to concentrate physical servers (blades) together. In the Smart Bank project, we concentrated 14 servers together, 12 Intel and 2 POWER4 blades. In the distributed world, most times, one server is dedicated to one application, thus isolating the environment. Following are some of the reasons for this:

- Facilitate troubleshooting
- Performance of the application
- Operating system compatibility with the application

The individual blade addresses these requirements and concentrates the physical implementation. The BladeCenter helps enterprises address hardware manageability, reliability, and flexibility by allowing clients to horizontally scale their implementation easily with different types of CPU, common chassis specifications, and hot swap capabilities, among other capabilities.

We deployed our additional branch server applications to four HS20 8832 G1X blades running SUSE Linux V9 on the BladeCenter in a WebSphere V6 cluster for high availability.

The z/VM environment

Our choice of mission-critical branch server implementation platform is the z/VM environment, where, unlike the BladeCenter, resources such as memory and CPU can be shared. Consequently, if one branch server is under utilized, we do not have resources tied specifically to that server. The available z990 or System z9-109 resources can be allocated to the branch servers that receive the bulk of the workload. One of our goals for Smart Bank is to show our clients that System z9-109 and IBM System z can always run at 100% CPU utilization. The virtual machines running on our z/VM LPAR on z990 (development environment for this book) and System z9-109 (production environment) run the same level of SUSE Linux V9 and the same WebSphere V6 cluster configuration, with the same branch server application as the BladeCenter.

Some of the other qualities of service that the z/VM environment brings to our branch servers are:

- Virtual network within the z9-109 machine using virtual local area network (VLAN) and HiperSockets
- Resiliency of the platform, with z/VM benefiting from some of the Geographically Dispersed Parallel Sysplex (GDPS) business continuity aspects, specifically, the GDPS/Peer-to-Peer Remote Copy (PPRC) HyperSwap of z/VM Linux to secondary disks in the event of disk failure
- Workload management on the z9-109 machine at the individual Linux guest machine level
- Virtualization capabilities of the z/VM environment, allowing considerable horizontal scalability and quick deployment of the virtual servers

Our IBM Tivoli Monitoring V6.1 views

With IBM Tivoli Monitoring V6.1, we monitored both these environments with the same Linux and WebSphere agents. This provided us with a view of our branch environment.

The BladeCenter solution can be viewed in our context as providing servers for specialized business functions that use the same core branch server functionality.

5.3.1 The Branch view

Figure 5-16 shows the detailed branch view of the workload from an operational point of view. In workspace A, the response time, in milliseconds, of the servlet receiving the workload on each machine, BladeCenter, and z/VM is seen. This response time reflects the routing shown in the JPEG diagram on the left, with the transaction being sent to our core system running in CICS with DB2 for z/OS, and the response returning to our Teller and Client Service Representative in the branch.

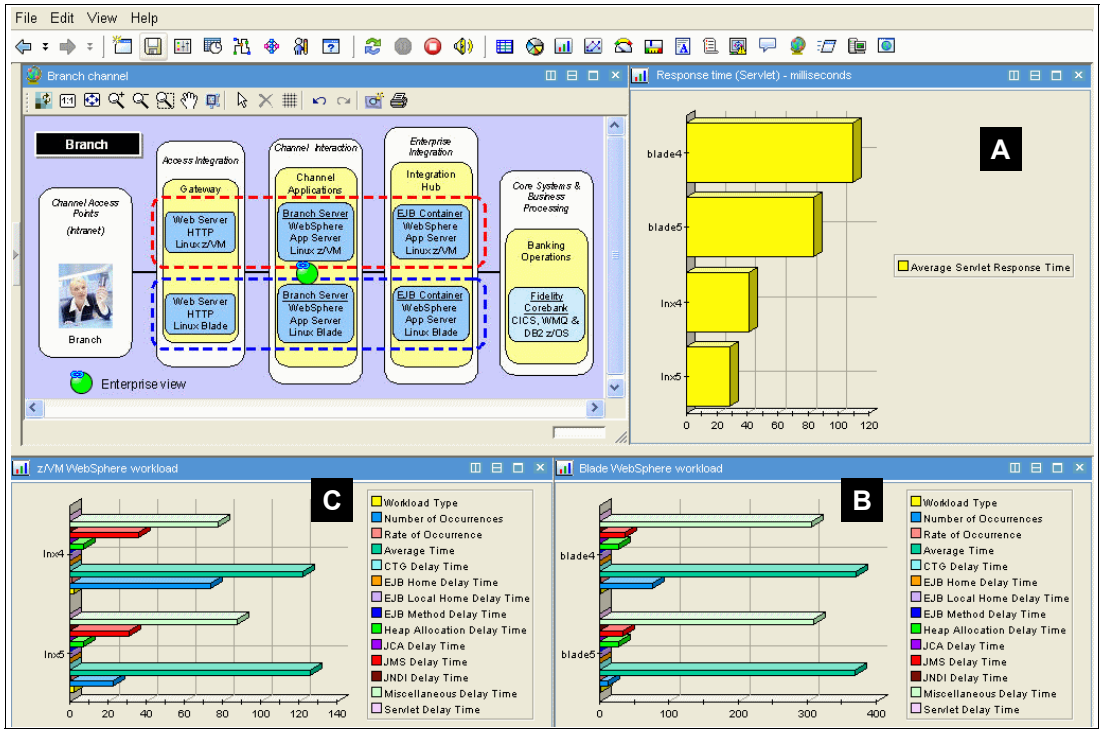


Figure 5-16 Branch view

Table 5-11 shows the base queries we used to create the Logical view from the assigned agents in Figure 5-16.

Table 5-11 Branch view queries

Workspace	Assigned agent and situation	Assigned managed system
A	WebSphere Application Servers for all versions/Web Applications/Query: *Web Applications*	WebSphere Application Server branch machines, lnx4 and lnx5 on z/VM and Blade4 and Blade5
B, C	WebSphere Application Servers for all versions/all workloads V2/query: *All Workloads*	WebSphere Application Server branch machines, lnx4 and lnx5 on z/VM and Blade4 and Blade5
Alert	Enterprise view	Link back to the Enterprise view
Alert	In the centre of the diagram	Link to the Linux utilization

Figure 5-16 clearly shows that lnx4 and Blade4 are receiving more workload that the other machines, which explains why they are experiencing slightly longer response times than the other application servers in their clusters, lnx5 and Blade5, respectively.

Using the alert in the center of the diagram, we can link to a Linux utilization Logical view, which then shows us, from a Linux perspective, what the physical blades and the virtual machines are doing.

5.3.2 The Linux utilization view

In the window shown in Figure 5-17, we created a Logical view of our branch server operating environment with our WebSphere clusters on z/VM and on BladeCenter. In both instances, we did not stress on the CPU resources available to us (see A and B). A view of the memory utilization in C and D is also available.

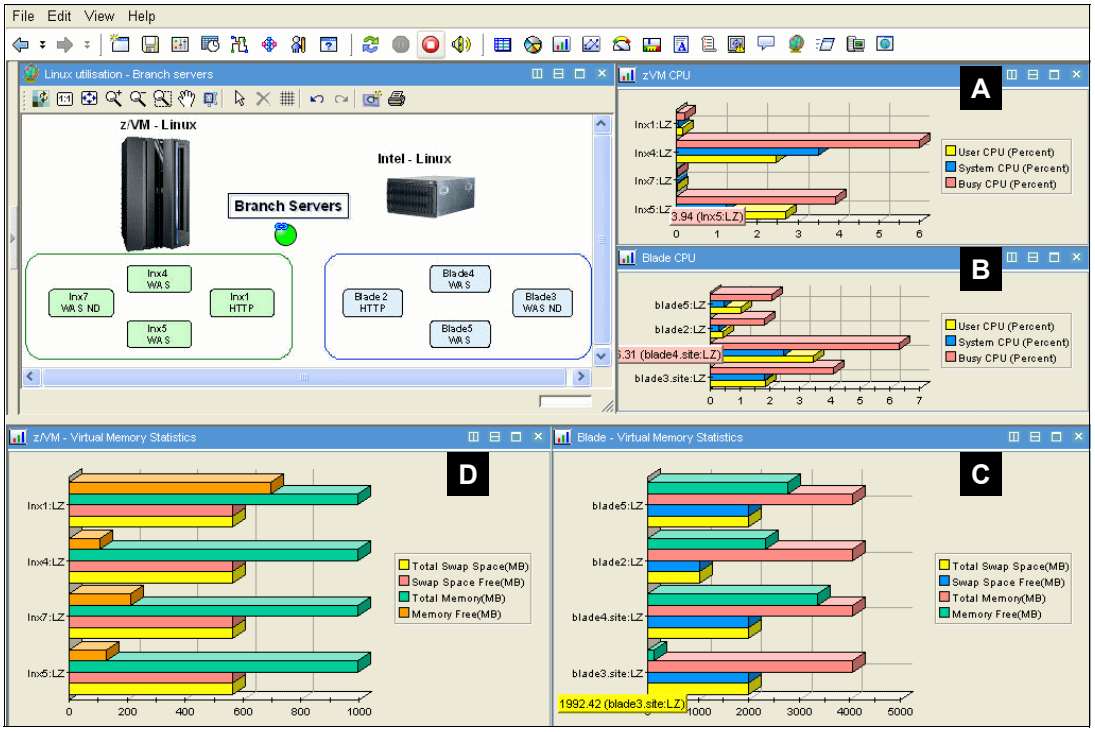


Figure 5-17 Linux environment

Table 5-12 shows the base queries used to create the Logical view from the assigned agents in Figure 5-16.

Table 5-12 Linux utilization

Workspace	Assigned agent and situation	Assigned managed system
A,B	Linux OS/Linux CPU/Query: *CPU*	WebSphere Application Server cluster machines on z/VM and BladeCenter
C,D	Linux OS/Linux VM Stats/Query: *VM Stats*	WebSphere Application Server cluster machines on z/VM and BladeCenter

Neither the BladeCenter nor the z/VM machine use virtual memory at this level of workload. We injected around 4 - 6 operations per second over the branch channel to the z990 (in our development region) when we captured these windows, with the other workloads running for ATM, Internet, and Batch.

It is important to monitor memory utilization. In the Linux environment, the virtual memory can be monitored through Total Swap Space (virtual memory defined to the system) and Swap Space Free. By managing the available system memory, both Random Access Memory (RAM) and virtual memory (Swap Space) help identify processes that are being constrained, not necessarily by the CPU, but by the available memory that may cause response time problems.

5.3.3 Summary

With the branch server Logical views, we leveraged the capabilities of IBM Tivoli Monitoring V6 to monitor and manage the heterogeneous environment. In the context of the Smart Bank showcase, we showed our IT simplification strategy at work and the overall monitoring capability with regard to our enterprise.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 286. Note that some of the documents referenced here may be available in soft copy only.

- *Deployment Guide Series: IBM Tivoli Monitoring 6.1*, SG24-7188
- *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143
- *IBM Tivoli Composite Application Manager V6.0 Family: Installation, Configuration, and Basic Usage*, SG24-7151
- *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155

Other publications

The following publications are also relevant as further sources of information:

- *CICS Business Event Publisher for MQSeries V1.2 Getting Started*, GC34-6296-002
- *IBM Tivoli Composite Application Manager for CICS Transactions Product Guide*, SC32-9510
- *IBM Tivoli Composite Application Manager for Response Time Tracking V6.0 Installation and Configuration Guide*, GC32-9482
- *IBM Tivoli Composite Application Manager for WebSphere Installation and Customization Guide*, GC32-9506
- *IBM Tivoli Composite Application Manager for WebSphere Operator’s Guide*, SC32-9508
- *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407
- *IBM Tivoli Monitoring Agent User’s Guide*, SC32-9459

Online resources

The soft copy of these publications can be downloaded from the following Web site:

`http://www-306.ibm.com/software/tivoli/library/`

The IBM Publications Center

`http://www.ibm.com/shop/publications/order`

For current information about IBM Tivoli software education, visit the IBM Tivoli software education Web page at:

`http://www-3.ibm.com/software/tivoli/education`

© Copyright IBM Corp. 2006. All rights reserved.

The current schedule of information about IBM Tivoli training is available on the IBM Tivoli software education schedules Web page at:

http://www-3.ibm.com/software/tivoli/education/schedules

The following Web sites and URLs are also relevant as further sources of information:

- Detailed information about Distributed Monitoring and OMEGAMON

  http://www-306.ibm.com/software/tivoli/sw-atoz/indexM.html

- Details pertaining to Request for Comments (RFC) error codes

  http://rfc.net/rfc2616.html

- IBM Tivoli Open Process Automation Library

  http://www-18.lotus.com/wps/portal/tm

- Latest documents pertaining to IBM Tivoli Monitoring

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.itm.doc/itm610usersguide.pdf

- LogFormat description and directives

  http://httpd.apache.org/docs/2.0/mod/mod_log_config.html#logformat

- OMEGAMON V350/V360 agents application enabling Web site

  https://www14.software.ibm.com/webapp/iwm/web/reg/download.do?source=tmdl&S_PKG=dl&cp=UTF-8

- OMEGAMON XE agents download

  https://www14.software.ibm.com/webapp/iwm/web/reg/download.do?source=tmdl&S_PKG=dl&cp=UTF-8

- Tivoli products installation documents

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml

- Up-to-date information about the platform support matrix for IBM Tivoli Monitoring V6.1

  http://www-306.ibm.com/software/sysmgmt/products/support/Tivoli_Supported_Platforms.html

How to get IBM Redbooks

You can search for, view, or download IBM Redbooks, IBM Redpapers, Hints and Tips, draft publications, and Additional materials, as well as order hardcopy IBM Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A
ACID 16
advanced link 214
architecture 261
assigned agent 266–267
  logical view 266
ATM 52, 192, 258
Atomic Consistent Isolated and Durable, *See* ACID
attribute group 28, 122
  EXCEPTION_DETAIL 174
  WORKLOAD_DETAIL 175

B
Banking Data Warehouse, *See* BDW
Basel II
  compliance 13
  credit risk reporting 15
  report 15
BDW 4
BPEL 19, 270
branch server 10, 263, 280
business continuity 2, 21
Business Event Publisher, *See* CICS BEP
Business Process Execution Language, *See* BPEL

C
CBM 14
CICS BEP 20, 154
  activity 163
  component 158–159
  Dataspace Server 158
  event connecter 158
  event connector 156
  event monitor 169
  for MQSeries 154
  key features 154
  Messaging Server 158
  plan 169
  process 161
  product 269
  setup process 158
CICS region 130, 192, 213
  current affectation 192
CICS Transaction Gateway 262–263
  mechanism 264
Component Business Model, *See* CBM
core system 4, 10, 258, 262
  API 261
  proprietary mechanism 261
  service-oriented invocation 262
  transformation 18, 257
CRB.SRVI NQ 269

D
DB2 55, 84, 94, 265, 279, 282
demonstration proof points 10
derived attribute 176
dynamic link
  context 224
  definition 215
    window 222, 240
  utilization 231

E
Early Support Programme, *See* ESP
EJB 22, 139, 263
  report activity level 139
Enterprise JavaBeans, *See* EJB
enterprise view 17, 203, 258
ESP 4
ETL 19
Extract Transform and Load, *See* ETL

F
Fair Isaac
  TRIAD 17
Fidelity Corebank 4
  application 265
Financial Services Architecture, *See* FSA
Financial Services Data Model, *See* FSDM
financial services sector 4
FSA 14
FSDM 4
  physical implementation 4

G
garbage collection 139
GDPS 21, 281
Geographically Dispersed Parallel Sysplex, *See* GDPS
Global Location Broker 82

H
Hardware Management Console, *See* HMC
HMC 272
HTTP session 140

I
IBM AIX 17, 53, 69, 125, 274
IBM BladeCenter 18, 278
  SUSE Linux V9 281
IBM OMEGAMON 124
  navigation subtree 128
  physical tree 149
IBM Redbooks Web site 286

© Copyright IBM Corp. 2006. All rights reserved.

IBM System z 23
IBM System z9
environment 258, 278
platform 271
utilization 278
IBM System z9-109
machine 21
IBM Tivoli 3
IBM Tivoli Composite Application Management 24, 125, 260
IBM Tivoli Composite Application Manager V6 3
IBM Tivoli Monitoring 3, 121–122, 274
agent 24, 268
component 26
core part 40
infrastructure 26, 122
management infrastructure 49
physical tree 130
product 122
services foundation 26
solution 24
system 128
system administrator 124
system management infrastructure 35
user guide 141
view 281
Independent Solution Vendor, *See* ISV
infrastructure perspective 10
user platform 10
infrastructure solution 2
integration hub 266
interface technique 262
Internet channel 12, 261
IP communication 39
hyper socket 39
ISV 4

J

J2EE 56, 139, 261
Java Message Service, *See* JMS
Java Virtual Machine, *See* JVM
JMS 112, 149, 261
JVM 21

L

link
absolute link 203
anchor 240
relative link 211
wizard 205
Linux OS 274
local location broker 82
local remote Tivoli Enterprise Management Server Hub 33
logical partition, *See* LPAR
logical tree 195, 259
selected level 195
logical view 16, 139, 181, 258
final objective 182
relative link 211
tree structure 202
LPAR 17, 276

M

managed system 50, 150, 181, 266
management query 261
MDB 263
Message Driven Bean, *See* MDB
message server 155
moveable hub 38, 276
MQSeries message
queue 154
server 155
multichannel integration 269
multichannel transformation 10

N

navigation subtree 126
navigator
item 123
view 163, 184, 275
existing level 187
queue statistics tab 163

O

On/Off Capacity on Demand 13
Open Infrastructure Offering 272
open standard 271

P

physical blade 283
physical tree 122, 182
physical view 22, 121, 195, 258
posting table 156
process choreographer 265
process EJB 262
simple RMI/IIOP call 271
production network 86
programming model 280

Q

quality of service 12

R

Random Access Memory (RAM) 284
real-time update 19
reference architecture 14
response time 20, 135, 258
retail bank 4, 260
recognizable view 260
simplified logical view 260
router status 178

S

SAN 49

seeding operation 95
SERCLASS symbol 228
  dynamic association 230
server list 82
service-oriented architecture 9
service-oriented architecture, *See* SOA
Services Inventory/Query 266
Smart Bank
  client 185, 248
  demonstration 10
  enterprise view 248
  environment 180
  infrastructure 247
  logical view 250
    tree 248
  main menu 247
  network workspace 252
  node 180
  operating environment 2
  operations workspace 250
  project 22, 121, 280
    goal 22
  proof point 8
  root 246
  server 180
  showcase 1, 32, 148, 192, 202, 258
    .css 244
    architecture 19
    demonstration 258
    environment 24
    image 231
    infrastructure 154
    logical view 242
    menu 203
    objective 4
    operations workspace 250
    proof points 23
    rule 128
    style 242
    system 128
    Welcome menu 250
    workspace 252
  Tivoli Enterprise Portal administrator 188
  Welcome menu 247
SMS 131
SNMP data provider 177
SOA 18, 131, 257
  model
    diagram 20
    view 268
  response time 136
  service 133
  Web services 138
STG 4
Storage Area Network, *See* SAN
Storage Management Subsystem, *See* SMS
systems management 3, 250, 273
  showcase requirements 22
Systems Technology Group, *See* STG

T

target workspace 201
  internal content 223
TCP/IP communication 45
technical architecture 1
TEPS 25
Tivoli Data Warehouse 28
Tivoli Enterprise 20, 122
Tivoli Enterprise Management Server
  primary 33
Tivoli Enterprise Management Server Hub 25–26, 66
  Hot Standby configuration 36
  logical gateway 28
  move 41
  store 26
Tivoli Enterprise Management Server Remote 25
  port switching 41
Tivoli Enterprise Managing Agent 25, 125, 189
Tivoli Enterprise Managing Server 25–26
Tivoli Enterprise Portal 12, 182, 258
Tivoli Enterprise Portal Client 122
Tivoli Enterprise Portal client 25
Tivoli Enterprise Portal Server, *See* TEPS

U

Universal Agent 125, 171, 274
use case 17, 257

W

Web service 31
  monitoring agent 31
Web Services Definition Language, *See* WSDL
WebSphere Application Server 17, 124, 261, 282
  agent 274
  deep-dive analysis 31
  defined data source 145
  IBM OMEGAMON 124
  IBM Tivoli Composite Application Manager 139
  instance 273
    workspace 142
  resource use 139
WebSphere Business Integration 125
  server foundation 18
WebSphere MQ 18, 132, 262
  activity 269
  IBM OMEGAMON 132
  logical view 268
  monitoring 149
  product 150
  queue 269
  system 150
WLM 21, 258
  service class
    resource 218
Work Load Manager, *See* WLM
workspace administration
  mode 182
WSDL 18, 262

Z

z/OS LPARs
    configuration values 125

Infrastructure Solutions: Building a Smart Bank Operating Environment

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Infrastructure Solutions: Building a Smart Bank Operating Environment

Using a live showcase to demonstrate how to deliver key banking projects

Building an SOA with IBM System z and IBM BladeCenter with IBM z/OS and Linux

Using IBM Tivoli Monitoring V6 and Tivoli Composite Application Manager V6

IBM created the Smart Bank showcase to demonstrate to retail banking clients the benefits of IBM products, services, and solutions in addressing their key business issues. The Smart Bank showcase covers different solutions areas such as IT service management, service-oriented architecture, optimization of IT resources, information lifecycle management, information warehouse, and business continuity. The Smart Bank showcase is an implementation of these solutions in a real environment.

This IBM Redbooks publication looks at the infrastructure monitoring and infrastructure management provided by IBM Tivoli Monitoring V6 and IBM Tivoli Composite Application Manager V6.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7113-00 ISBN 0738496804